Documentation 8.2

ZABBIX

05.11.2026

Contents

Руководство пользователя **5**

Добро пожаловать в документацию Zabbix **5**

Начало работы с Zabbix . . . 5
Zabbix Cloud . . . 5
Мобильное приложение . . . 5
Центр разработчиков . . . 5
Сообщество и другие ресурсы . . . 5
Установка и первые шаги . . . 6
Установка веб-интерфейса . . . 6
Краткие руководства . . . 11
Установка: дополнительные руководства . . . 85
Обновление . . . 168
Известные проблемы . . . 185
Требования . . . 194
Новое и примечания по обновлению: 8.0 . . . 204
Примечания к обновлению . . . 204
Изменения, несовместимые с предыдущими версиями . . . 205
Другое . . . 205
Новые возможности и примечания по обновлению: 8.0.x . . . 205
Zabbix appliance . . . 205
Процессы . . . 209
Сервер . . . 211
Прокси . . . 239
Агент . . . 286
Агент 2 . . . 318
Java gateway . . . 383
Отправитель . . . 388
Получить . . . 390
JS . . . 391
Веб-сервис . . . 392
MCP-сервер . . . 395
Адаптер моста . . . 403
Переменные среды . . . 407
Конфигурация . . . 409
Узлы сети и группы узлов сети . . . 420
Элементы данных . . . 439
Обнаружение проблем с помощью триггеров . . . 673
Визуализация . . . 727
Шаблоны и группы шаблонов . . . 754
Шаблоны из коробки . . . 754
Уведомления при событиях . . . 768
Макросы . . . 822
Пользователи и группы пользователей . . . 837
Запланированные отчеты . . . 848
Экспорт данных . . . 852
Экспорт/импорт конфигурации . . . 861
Регулярные выражения . . . 928
События и разрешение . . . 935
Подтверждение и подавление проблем . . . 937
Ручное закрытие проблем . . . 941

Периоды обслуживания . . . 942
Тегирование . . . 948
Complex event processing . . . 951
Корреляция событий . . . 955
Обнаружение . . . 961
Обнаружение сети . . . 962
Автоматическая регистрация активного агента . . . 971
Обнаружение на низком уровне . . . 975
Мониторинг виртуальных машин . . . 1000
Ключи элементов данных для мониторинга VMware . . . 1001
Поля ключей обнаружения виртуальных машин . . . 1020
Примеры JSON для элементов данных VMware . . . 1024
Пример настройки мониторинга VMware . . . 1029
Веб-мониторинг . . . 1034
Элементы данных веб-мониторинга . . . 1043
Реальный сценарий . . . 1044
Мониторинг сервисов . . . 1051
Дерево услуг . . . 1052
SLA . . . 1056
Пример настройки . . . 1058
Обновление мониторинга сервисов . . . 1062
Шифрование и безопасность . . . 1063
Шифрование с использованием сертификатов . . . 1072
Шифрование с использованием предварительно общих ключей . . . 1079
Безопасное подключение к базе данных . . . 1082
Защищенное соединение между веб-интерфейсом и сервером . . . 1095
Управление доступом . . . 1099
Криптография . . . 1103
Защита веб-сервера . . . 1104
Хранение секретов . . . 1108
Доверие к TLS-сертификату Google Chrome . . . 1116
Лучшие практики безопасности . . . 1118
Веб-интерфейс . . . 1118
Меню . . . 1119
Разделы веб-интерфейса . . . 1129
Настройки пользователя . . . 1413
Глобальный поиск . . . 1419
Режим обслуживания веб-интерфейса . . . 1421
Параметры страницы . . . 1422
Определения . . . 1424
Дополнительные языки веб-интерфейса . . . 1425
Создание собственной темы . . . 1426
Режим отладки . . . 1427
Файлы cookie, используемые Zabbix . . . 1428
Часовые пояса . . . 1429
Сброс пароля . . . 1430
Период времени и селекторы узлов сети . . . 1430
Параметры конфигурации: веб-интерфейс Zabbix . . . 1432
API . . . 1439
Справочник методов . . . 1444
Приложение 1. Справочные комментарии . . . 2350
Приложение 2. Изменения с 7.4 до 8.0 . . . 2358
Приложение 3. Изменения в 8.0 . . . 2358
Расширения . . . 2358
Загружаемые модули . . . 2360
Модули веб-интерфейса . . . 2369
Приложения . . . 2370
Установка и настройка . . . 2371
Элементы данных . . . 2418
Поддерживаемые функции . . . 2460
Макросы . . . 2460
Совместимость версий . . . 2497
Синтаксис временного периода . . . 2499

Обозначения единиц измерения . . . 2499
Выполнение команд . . . 2501
Примеры экранирования . . . 2502
Что такое Zabbix . . . 2503

Zabbix Cloud **2508**

Zabbix Cloud **2508**
Настройка и управление Zabbix Cloud . . . 2508
Узнайте о Zabbix Cloud . . . 2508
Развертывание Zabbix в облаке . . . 2508
Конфигурация узла сети . . . 2518
Обзор . . . 2519
Вкладка «Обзор» . . . 2520
Вкладка «Фильтры доступа» . . . 2523
Вкладка «Шифрование» . . . 2524
Вкладка «Резервные копии» . . . 2524
Вкладка «История» . . . 2525
Вкладка «Обслуживание» . . . 2526
Вкладка «Обновление» . . . 2527
Добавление пользователей . . . 2528
Введение . . . 2528
Добавление пользователей в организации . . . 2528
Добавление пользователей к нодам . . . 2530
Ключевые отличия Zabbix Cloud . . . 2530
Введение . . . 2530
Журнал аудита . . . 2533
Настройка облачного прокси Zabbix . . . 2535

Мобильное приложение **2543**

Мобильное приложение **2543**
Настройка Zabbix для приложения . . . 2543
Узнайте о приложении . . . 2544
Настройка Zabbix для приложения . . . 2544
Обзор приложения . . . 2545
Безопасный мобильный доступ . . . 2553

Центр разработчика **2554**

Центр разработчика **2554**
Сборка на Zabbix . . . 2554
Создание плагина . . . 2555
Интерфейсы плагинов . . . 2560
Библиотека Python для Zabbix . . . 2562
Установка . . . 2562
Руководство по быстрому началу работы . . . 2563
Использование Zabbix API . . . 2566
Сбор данных с агента Zabbix . . . 2569
Отправка данных на сервер Zabbix или прокси . . . 2571
Отладочное логирование . . . 2574
Модули . . . 2575
Файловая структура модуля . . . 2576
Виджеты . . . 2583
Руководства . . . 2592
Примеры . . . 2618
Изменения в разработке расширений . . . 2619

Страницы помощи Zabbix **2619**
zabbix_agent2 . . . 2619
zabbix_agentd . . . 2620
zabbix_get . . . 2622
zabbix_js . . . 2624
zabbix_proxy . . . 2625
zabbix_sender . . . 2627

zabbix_server 2632
zabbix_web_service 2635

Уведомление об авторских правах **2635**

Руководство пользователя

Добро пожаловать в документацию Zabbix

Ваш основной источник информации по работе с мониторингом Zabbix — от базовой настройки до расширенных конфигураций. Это руководство охватывает все необходимое для установки, настройки и эксплуатации Zabbix.

Что нового в Zabbix 8.0

Начало работы с Zabbix

Установка Пошаговые инструкции по установке Zabbix на предпочитаемой вами платформе, включая различные операционные системы и конфигурации баз данных.

Требования Список поддерживаемых платформ и необходимых программных компонентов, чтобы помочь вам подготовить среду для успешного развёртывания Zabbix.

Краткие руководства Краткие практические инструкции, которые проведут вас через основы — от первых шагов по настройке до получения первого уведомления о проблеме.

Zabbix Cloud

Начните использовать Zabbix Cloud Пошаговые инструкции по быстрой настройке и запуску вашего экземпляра Zabbix Cloud, включая первоначальную настройку и подключение отслеживаемых устройств.

Zabbix: облачная версия или локальная Сравните различия в управлении, масштабируемости и обслуживании между двумя вариантами развёртывания и выясните, какой из них лучше всего подходит для поддержки вашего рабочего процесса.

Изучите Zabbix Cloud Ознакомьтесь с обзором Zabbix Cloud, включая его ключевые функции, преимущества и условия подписки. Часто задаваемые вопросы по настройке и использованию смотрите в разделе FAQ.

Мобильное приложение

Получить приложение Установите мобильное приложение Zabbix (скоро будет доступно) на свое устройство и начните мониторить среду Zabbix в любое время.

Настройка Zabbix для приложения Установите адаптер-мост Zabbix, чтобы мобильные устройства могли подключаться к Zabbix. Затем добавьте свое устройство и включите push-уведомления.

Обзор приложения Изучите основы использования мобильного приложения Zabbix.

Центр разработчиков

Модули веб-интерфейса Руководства и справочные материалы по созданию пользовательских модулей, расширяющих или модифицирующих веб-интерфейс Zabbix для решения конкретных задач.

Виджеты Подробный анализ структуры и логики виджетов, а также инструкции по созданию пользовательских элементов панели, адаптированных под ваши потребности.

Плагины Обзор того, как разрабатывать и управлять плагинами Zabbix для расширения функциональности или интеграции с внешними системами.

Сообщество и другие ресурсы

Форумы Zabbix Пространство для обмена идеями, поиска решений и обмена опытом между специалистами Zabbix всех уровней.

Блог Zabbix Новости, обучающие материалы и примеры из практики, подготовленные командой Zabbix и участниками сообщества.

Видеоинструкции Видеотека с демонстрациями, обсуждениями и советами, которые помогут вам максимально эффективно использовать Zabbix.

Установка и первые шаги

Следующая последовательность описывает процесс настройки Zabbix:

1. Убедитесь, что необходимое программное обеспечение установлено и запущено.
2. Установите Zabbix.
3. Установите веб-интерфейс Zabbix.
4. Начните мониторинг.

Подробности по каждому шагу приведены ниже.

Необходимое программное обеспечение Zabbix хорошо работает со следующим программным обеспечением (часто называемым стеком LAMP):

- **Операционная система Linux:** Ubuntu, Debian, CentOS или другие
- **Веб-сервер:** Apache или NGINX
- **Сервер базы данных:** MySQL/MariaDB или PostgreSQL
- **PHP**

В большинстве современных систем Linux PHP уже может быть установлен. В этом случае может потребоваться только настроить сервер базы данных и веб-сервер.

Более подробную информацию, например о поддерживаемых версиях программного обеспечения, см. в разделе Необходимое программное обеспечение.

Установка Zabbix **Самый простой** способ установить Zabbix — использовать **пакеты дистрибутива** (альтернативный вариант — установка из контейнеров). Пакеты устанавливают необходимые программные зависимости.

Точные инструкции по установке для вашей операционной системы доступны на **веб-сайте Zabbix**.

Выберите свою платформу и следуйте приведенным инструкциям. Для начала рекомендуется установить сервер Zabbix, агент Zabbix 2 и веб-интерфейс. Позже можно добавить прокси Zabbix и другие агенты.

Часть процесса установки включает создание базы данных (для хранения данных конфигурации и истории) и импорт в нее исходных данных.

Установка веб-интерфейса Zabbix Выполните шаги **мастера установки** веб-интерфейса.

Вам потребуется указать пароль базы данных (созданный на предыдущем шаге), чтобы веб-интерфейс мог взаимодействовать с базой данных.

Начало мониторинга Войдите в веб-интерфейс Zabbix.

Используйте **краткие руководства**, чтобы начать мониторинг популярных ресурсов (Linux, Apache, MySQL и других). В этих руководствах объясняется, с какими шаблонами следует работать для начала мониторинга популярных ресурсов.

Можно начать с краткого руководства по мониторингу сервера Linux.

Установка веб-интерфейса

В этом разделе приведены инструкции по установке веб-интерфейса Zabbix. Выполните эти действия после установки серверной части Zabbix.

Веб-интерфейс Zabbix написан на PHP и требует поддерживаемый веб-сервер, например Apache или NGINX, с настроенным PHP. Информацию о необходимом программном обеспечении и библиотеках см. в разделе требований к веб-интерфейсу.

Чтобы использовать в веб-интерфейсе Zabbix язык, отличный от английского, необходимо установить его локаль на веб-сервере, а также расширение PHP gettext. Обратите внимание, что изменение языка может изменить формат времени и даты.

Чтобы настроить SSL для веб-интерфейса Zabbix, см. раздел Настройка SSL для веб-интерфейса Zabbix.

Мастер установки

Чтобы установить веб-интерфейс, откройте URL веб-интерфейса Zabbix в браузере:

- Для Apache: `http://<server_ip_or_name>/zabbix`

- Для NGINX: `http://<server_ip_or_name>`

Экран приветствия

Мастер установки веб-интерфейса открывается на экране приветствия.

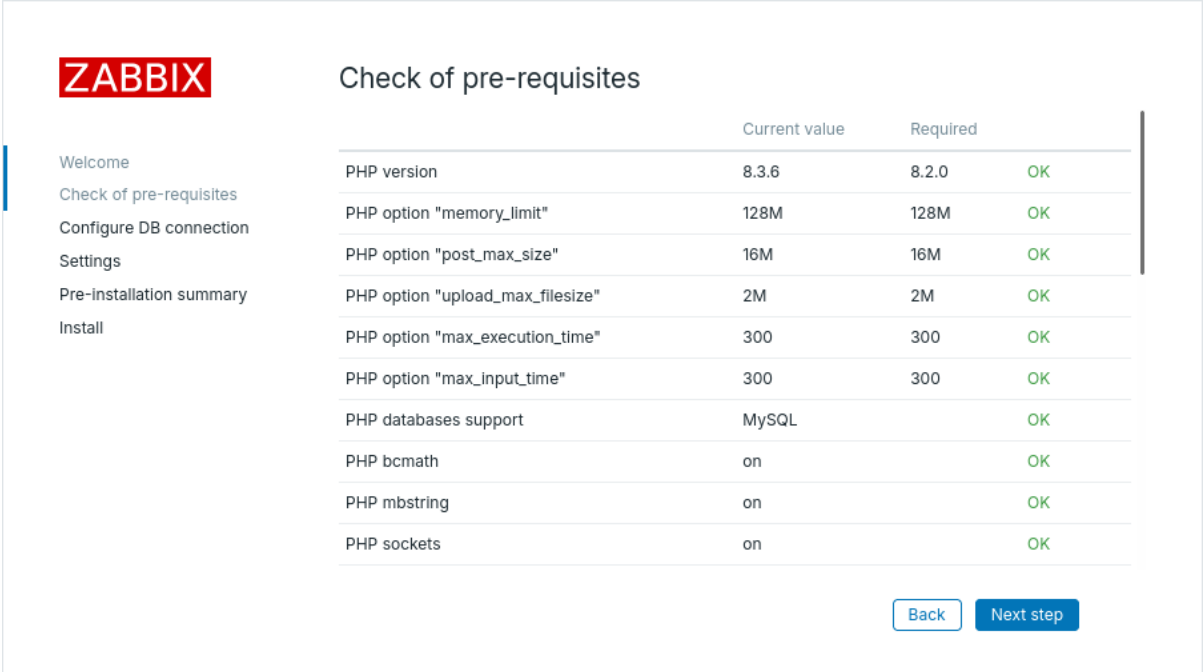


Системным языком по умолчанию является английский (en_US), а для времени и даты используется формат США. Чтобы изменить системный язык по умолчанию, выберите нужный язык в раскрывающемся списке *Язык по умолчанию*.

Нажмите кнопку *Следующий шаг*, чтобы продолжить.

Проверка предварительных требований

Убедитесь, что выполнены все обязательные предварительные требования для веб-интерфейса Zabbix.



Список необходимых и необязательных расширений PHP, требуемую версию PHP и поддерживаемые базы данных см. в разделе Требования. Обратите внимание, что невыполненные необязательные предварительные требования отображаются со статусом *Предупреждение*, и процесс настройки может продолжаться, даже если они не выполнены.

Требуемые значения директив конфигурации PHP в файле `php.ini`, используемом веб-сервером:

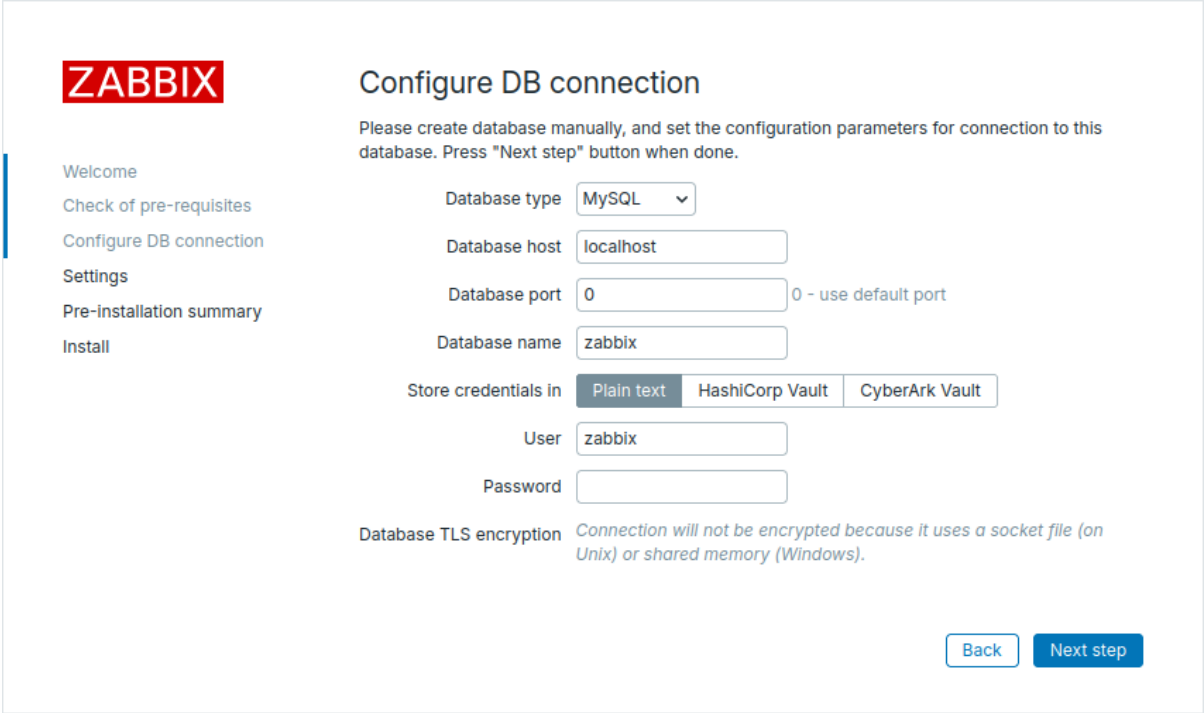
```
memory_limit = 128M ;
post_max_size = 16M ;
upload_max_filesize = 2M ;
max_execution_time = 300 ;             ;                  0   -1
max_input_time = 300 ;             ;                  0   -1
session.auto_start = 0 ;
arg_separator.output = "&" ;            "&amp;"
```

Attention:
Если необходимо изменить пользователя или группу веб-сервера, убедитесь, что новый пользователь или группа имеют необходимые разрешения для каталога сессий PHP; в противном случае настройка Zabbix может не продолжиться.

Если все обязательные предварительные требования выполнены, перейдите к следующему шагу.

Настройка подключения к базе данных

Введите сведения для подключения к базе данных.



Если поле *Узел сети базы данных* оставить пустым, будет использоваться Unix-сокет по умолчанию.

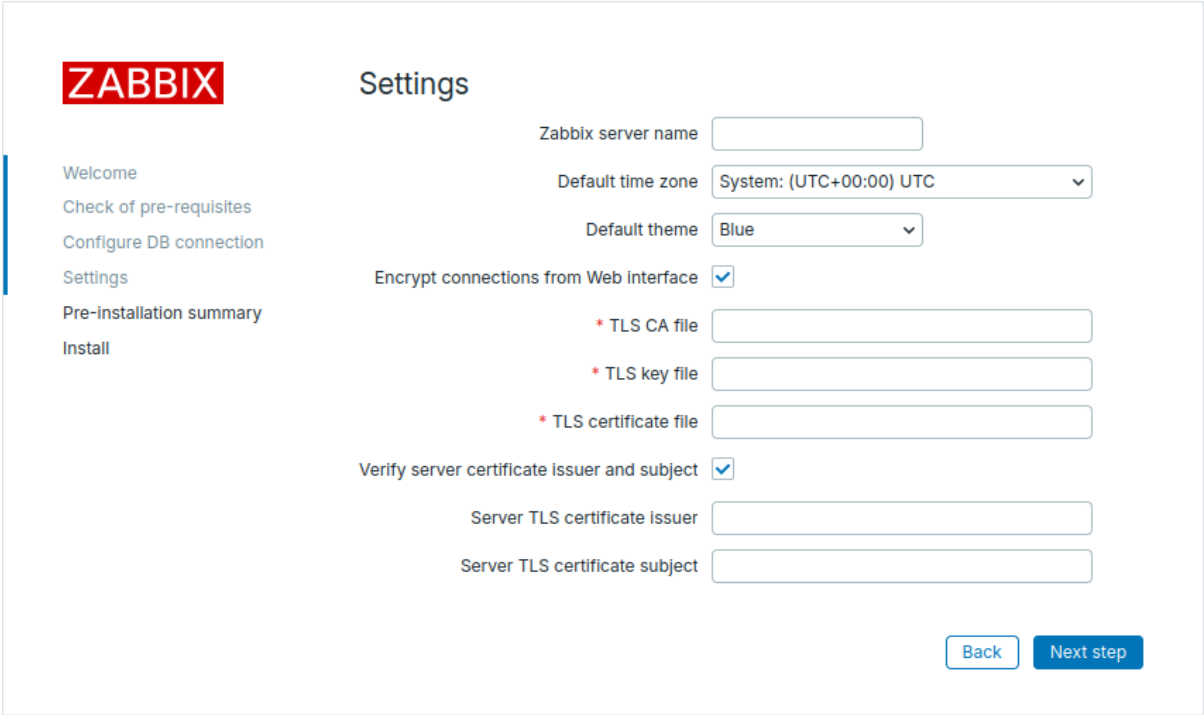
Для MySQL ввод значения `localhost` в поле *Узел сети базы данных* также приводит к подключению через Unix-сокет по умолчанию. Мастер настройки не предоставляет возможности настроить сокет базы данных. Чтобы использовать пользовательский сокет, настройте его в параметрах сервера Zabbix (например, с помощью параметра `DBSocket=` в файле конфигурации сервера Zabbix (`zabbix_server.conf`)). Это обеспечивает согласованность веб-интерфейса с настройками подключения сервера к базе данных.

Для PostgreSQL поле *Узел сети базы данных* может содержать несколько записей `host:port`, разделенных запятыми. Строка разбивается на части, и каждый порт проверяется так же, как в поле *Порт базы данных*. После успешной проверки строка разбирается на узлы сети и порты для формирования строки подключения. Если для узла сети в списке, разделенном запятыми, порт не указан, в качестве порта по умолчанию используется значение поля *Порт базы данных*. Если указан путь к сокету (например, `/var/run/pgbouncer`), вместо этого используется соответствующий Unix-сокет.

Чтобы настроить защищенное подключение к базе данных, установите флажок *Шифрование TLS подключения к базе данных* и перейдите к разделу Веб-интерфейс к базе данных, чтобы настроить поля. Флажок *Шифрование TLS подключения к базе данных* отключен в следующих случаях: - Для MySQL в поле *Узел сети базы данных* указано значение *localhost*. - Для PostgreSQL значение поля *Узел сети базы данных* начинается с косой черты или поле пусто.

Чтобы настроить Zabbix для получения секретов из HashiCorp Vault, выберите *HashiCorp Vault* в поле *Хранить учетные данные в* и перейдите к разделу Настройка HashiCorp, чтобы заполнить появившиеся поля. Чтобы настроить Zabbix для получения секретов из CyberArk Vault, выберите *CyberArk Vault* в поле *Хранить учетные данные в* и перейдите к разделу Настройка CyberArk, чтобы заполнить появившиеся поля.

Настройки



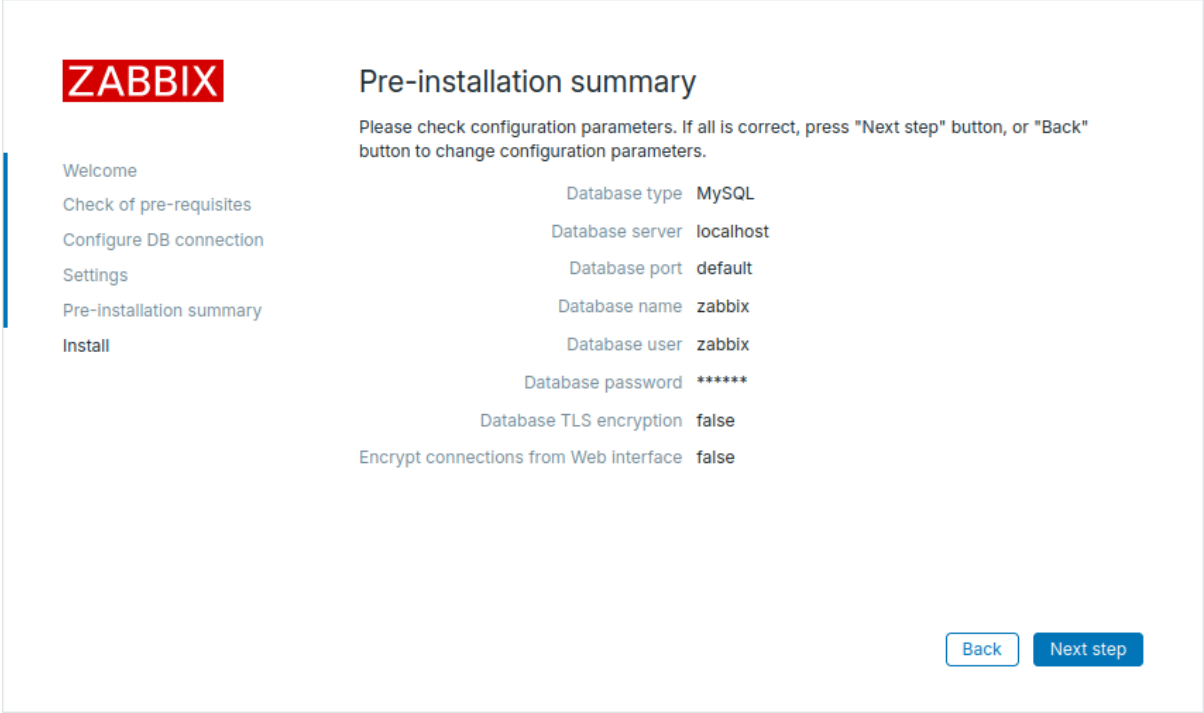
Ввод имени сервера Zabbix необязателен. Однако если имя указано, оно будет отображаться в строке меню и заголовках страниц.

Установите часовой пояс по умолчанию и тему для веб-интерфейса.

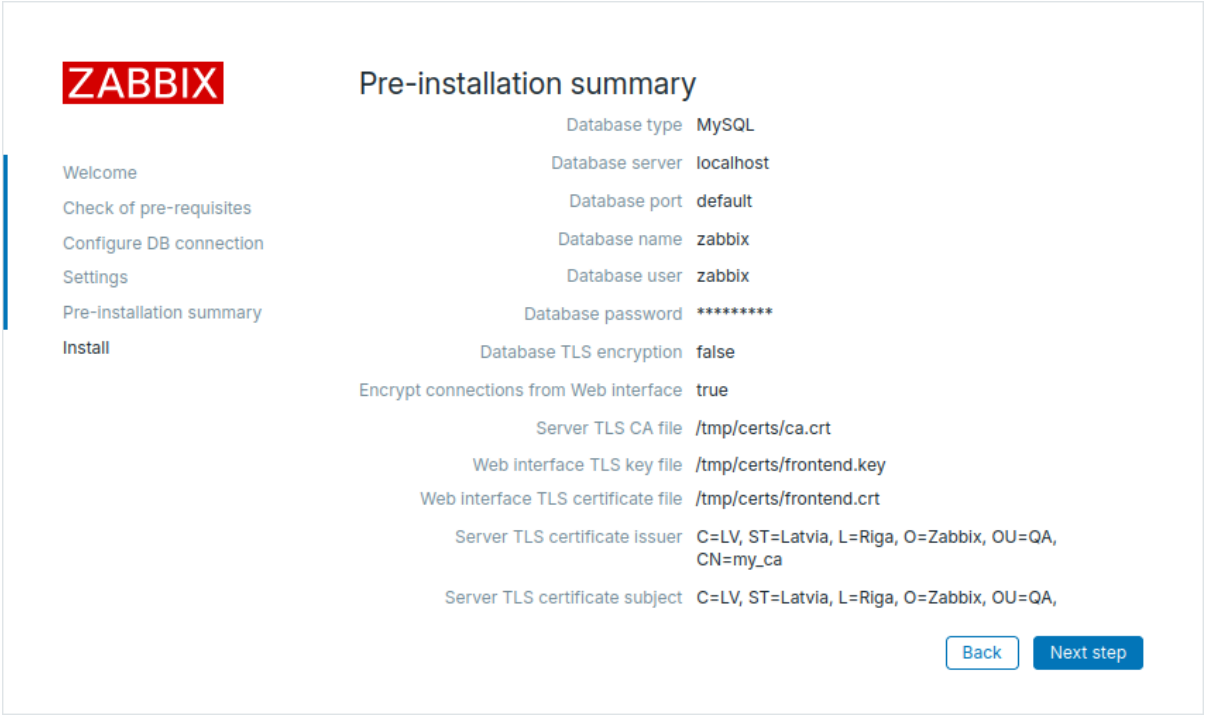
Чтобы настроить TLS-соединения между веб-интерфейсом Zabbix и сервером Zabbix, установите флажок *Шифровать соединения из веб-интерфейса* и перейдите к настройке TLS-соединения, чтобы заполнить появившиеся поля.

Сводка перед установкой

Просмотрите сводку настроек.



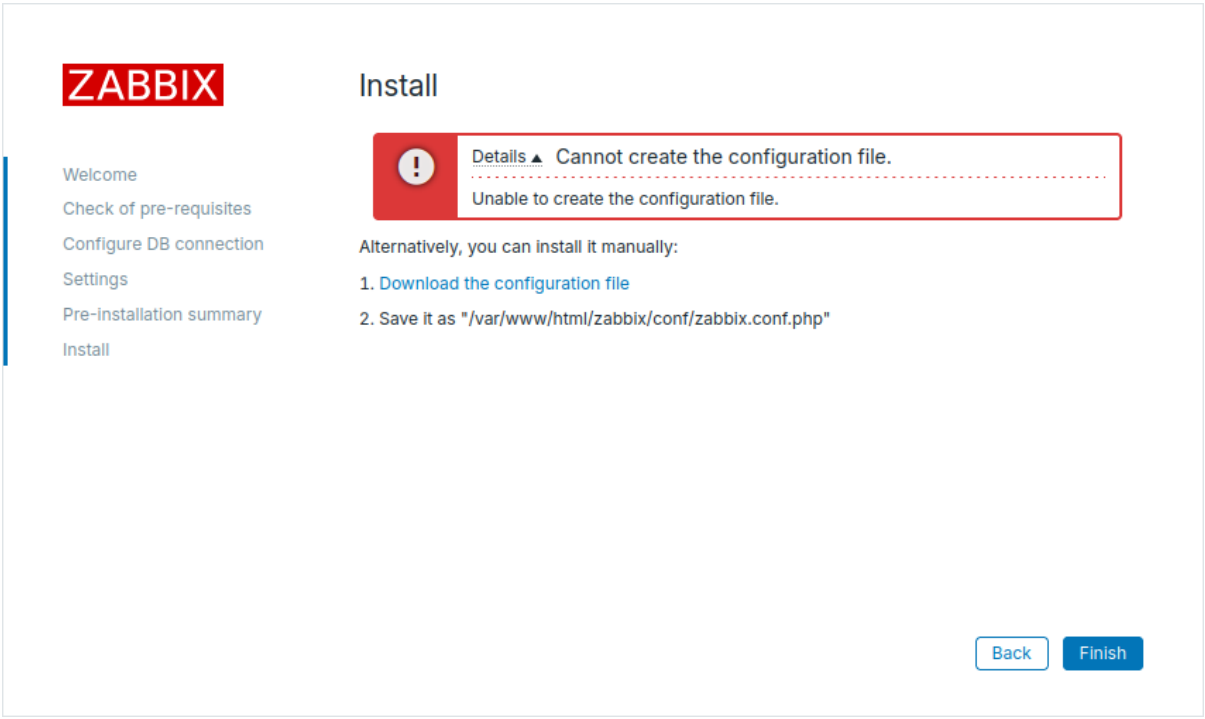
Если настроен TLS, мастер отобразит сведения о нем.



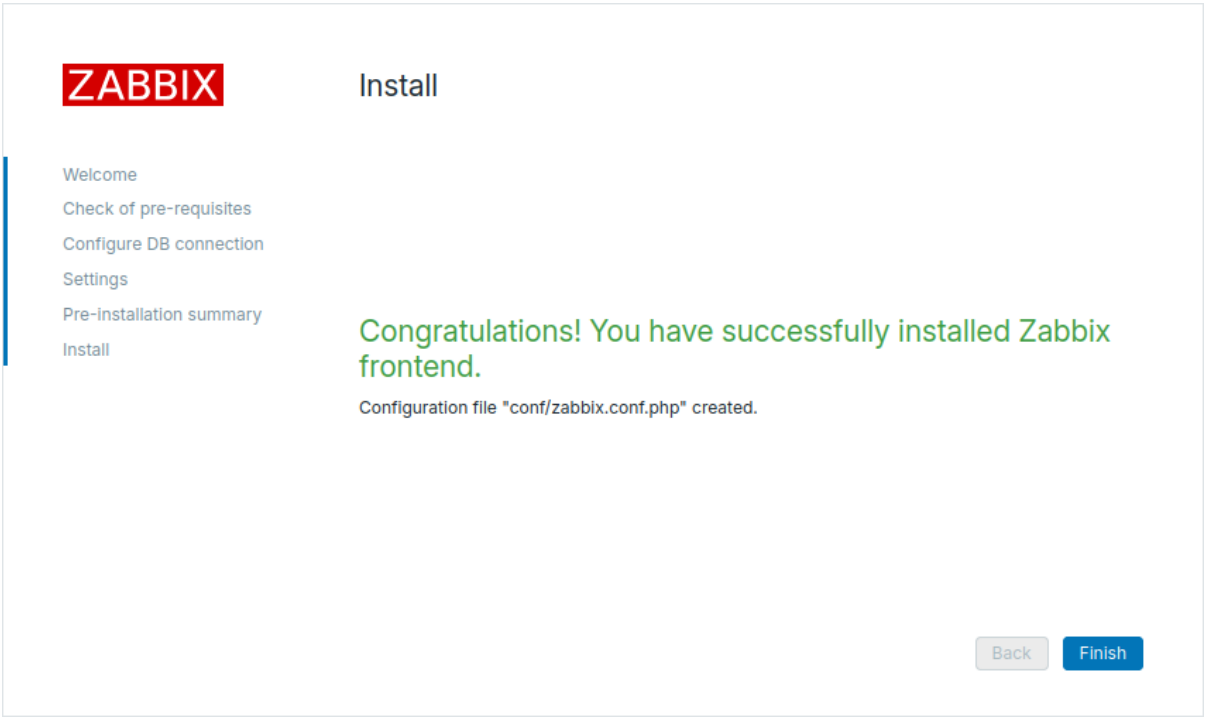
Установка

Если пользователь веб-сервера имеет доступ на запись в каталог `conf/`, файл конфигурации будет сохранен автоматически, и вы сможете сразу перейти к следующему шагу.

Если пользователь веб-сервера не имеет доступа на запись в каталог `conf/`, необходимо скачать файл конфигурации и поместить его в каталог `conf/` в каталоге HTML-документов веб-сервера, куда вы скопировали PHP-файлы Zabbix.

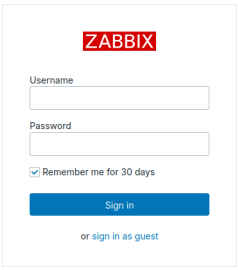


Завершите установку.



Вход

Войдите с учетными данными по умолчанию: - Имя пользователя: *Admin* - Пароль: *zabbix*



Файл конфигурации веб-интерфейса

После установки можно изменить конфигурацию веб-интерфейса Zabbix, обновив параметры в файле конфигурации веб-интерфейса `zabbix.conf.php`.

Путь к файлу конфигурации по умолчанию зависит от операционной системы и способа установки Zabbix:

- При установке из пакетов: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- При установке из исходных кодов: `/var/www/html/zabbix/conf/zabbix.conf.php`
- Для программно-аппаратного комплекса: `/etc/zabbix/web/zabbix.conf.php`

Подробнее о настройке веб-интерфейса Zabbix см. в разделе параметры файла конфигурации.

Краткие руководства

После того как вы установили Zabbix и его веб-интерфейс, можно приступать к мониторингу.

Используйте эти краткие руководства, чтобы начать мониторинг популярных ресурсов:

- Мониторинг сервера Linux
- Мониторинг веб-сервера Apache
- Мониторинг сервера MySQL

Краткие руководства посвящены использованию официальных шаблонов Zabbix для мониторинга.
Шаблон представляет собой предварительно настроенный набор элементов данных мониторинга.
Шаблон позволяет масштабировать мониторинг: вы можете протестировать шаблон на одном объекте мониторинга (устройстве, приложении и т. д.), настроить его, а затем применить к нескольким аналогичным объектам мониторинга.

Шаблоны подходят для мониторинга сразу после установки, однако вам также может потребоваться настроить их элементы данных.

Чтобы лучше понять основные элементы, из которых состоит шаблон, ознакомьтесь с разделом Базовая настройка.

Базовая конфигурация

Эти руководства знакомят с основными элементами конфигурации Zabbix:

- Вход в систему и настройка пользователя
- Новый узел сети
- Новый элемент данных
- Новый триггер
- Получение уведомления о проблеме
- Новый шаблон

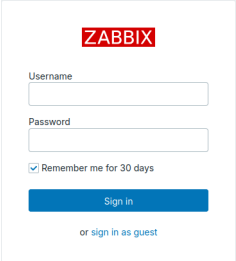
Вход в систему и настройка пользователя

Обзор

В этом разделе вы узнаете, как войти в систему и настроить системного пользователя в Zabbix.

Вход

Откройте URL веб-интерфейса Zabbix в браузере: - Для Apache: http://<server_ip_or_name>/zabbix - Для NGINX: http://<server_ip_or_name>



Это приветственный экран Zabbix. Введите имя пользователя **Admin** и пароль **zabbix**, чтобы войти как суперпользователь Zabbix. Будет предоставлен доступ ко всем разделам меню.

В целях безопасности настоятельно рекомендуется сразу после первого входа изменить пароль по умолчанию для учетной записи Admin.

Постоянный вход в систему

Чтобы оставаться в системе до 30 дней, выберите *Запомнить на 30 дней* перед нажатием *Войти*.

Если *Запомнить на 30 дней* включено:

- Ваша сессия остается активной в течение 30 дней.
- *Автовыход* переопределяется, и вы остаетесь в системе до окончания этого периода.
- При последующих посещениях в течение 30 дней вы будете входить автоматически без повторного ввода учетных данных.

Если *Запомнить на 30 дней* отключено:

- Любой ранее включенный автоматический вход очищается.
- Сессия истечет в соответствии с настроенным интервалом *Автовыход*.

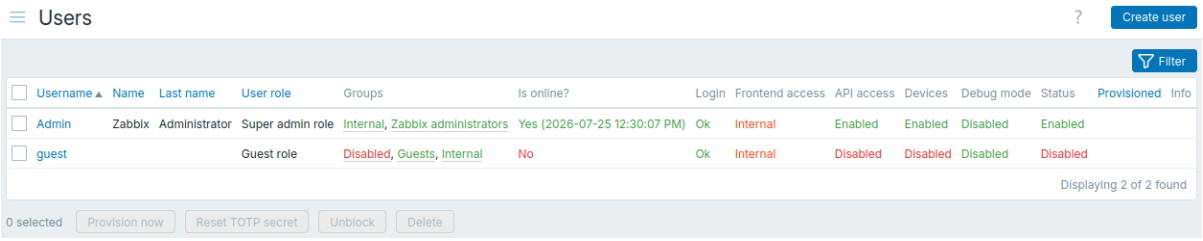
Защита от атак перебором

В случае пяти последовательных неудачных попыток входа веб-интерфейс Zabbix приостановит работу на 30 секунд, чтобы предотвратить атаки перебором и словарные атаки.

IP-адрес неудачной попытки входа будет отображаться после успешного входа.

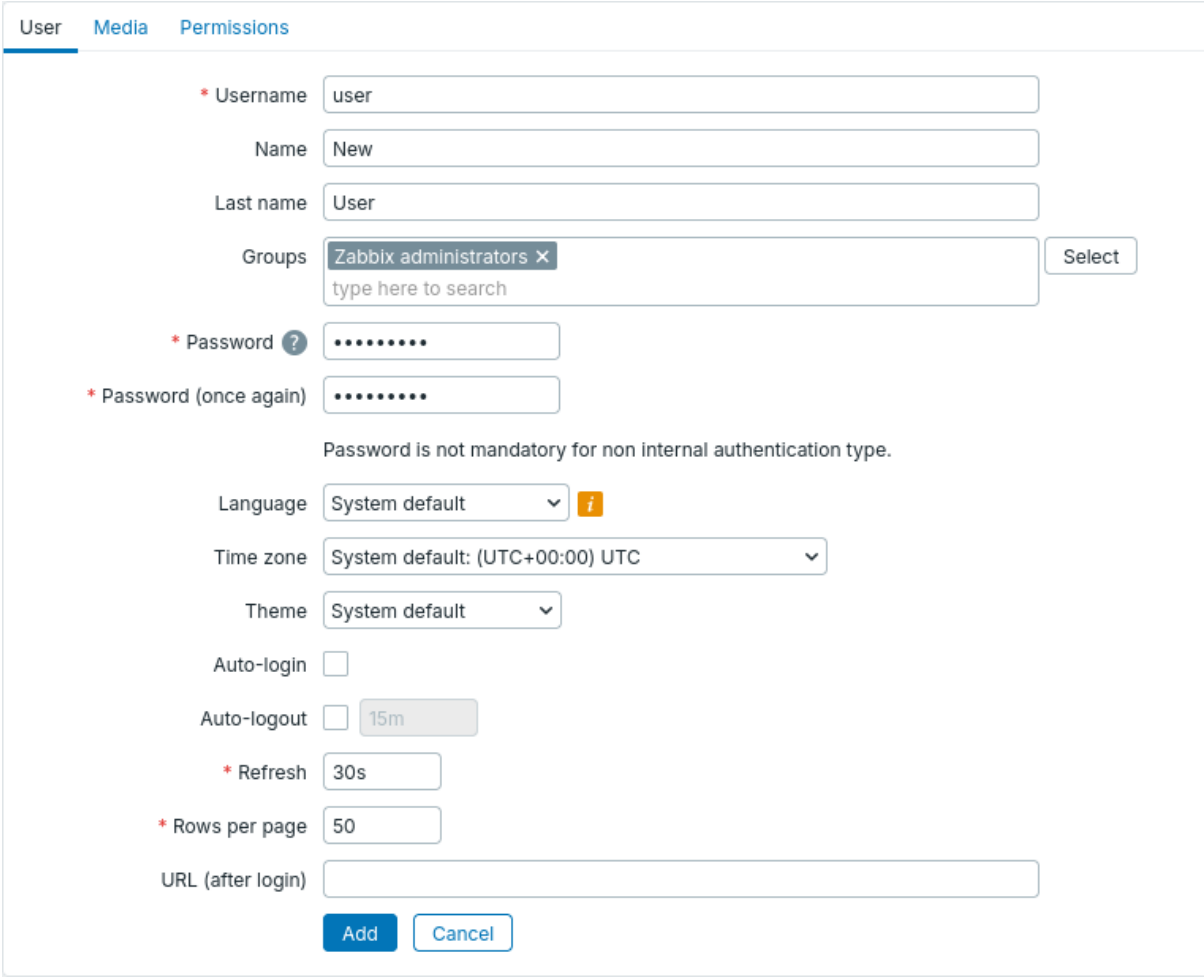
Добавление пользователя

Чтобы просмотреть информацию о пользователях, перейдите в *Пользователи* > *Пользователи* в вертикальном меню боковой панели.



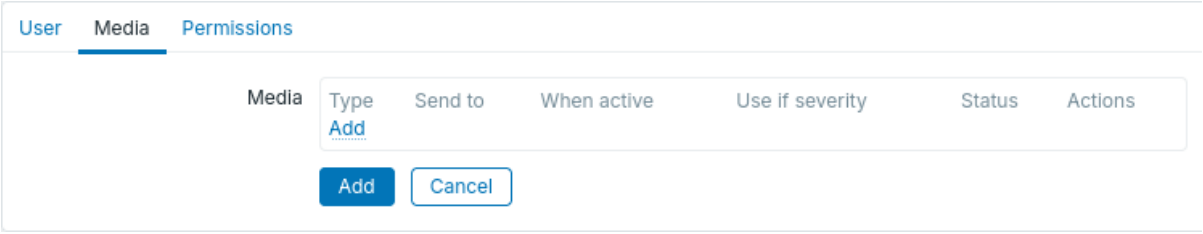
Чтобы добавить нового пользователя, выберите *Создать пользователя* в правом верхнем углу.

В форме нового пользователя обязательно добавьте пользователя в одну из существующих групп пользователей, например в 'Zabbix administrators'.



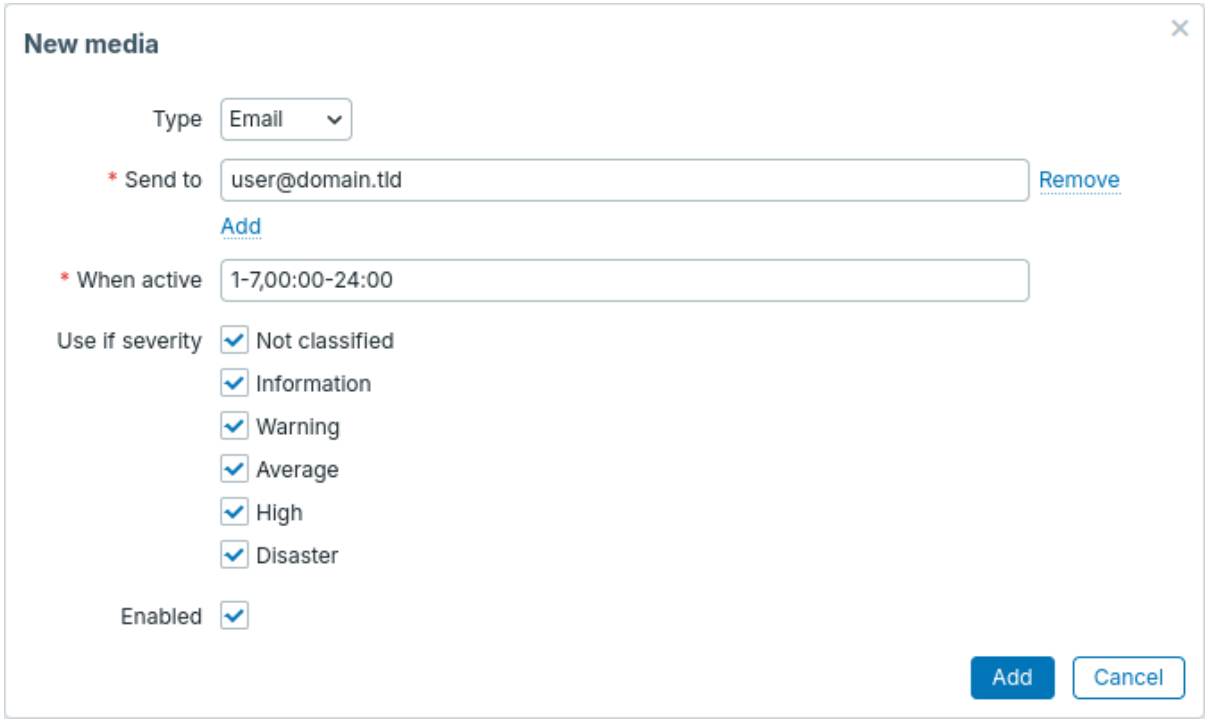
Все обязательные поля ввода отмечены красной звездочкой. Подробнее о полях ввода в этой форме настройки см. на странице Настройки пользователя.

По умолчанию для новых пользователей не задано ни одного способа доставки уведомлений (media). Чтобы создать его, перейдите на вкладку *Media* и нажмите *Add*.



Во всплывающем окне введите адрес электронной почты пользователя.
 Все типы media по умолчанию отключены и могут быть включены в *Уведомления* > *Типы media*.

Вы можете указать период времени, в течение которого medium будет активен (описание формата см. на странице Спецификация временного периода). По умолчанию medium всегда активен. Вы также можете настроить уровни серьезности триггера, для которых medium будет активен, но пока оставьте все включенными.



Нажмите *Add*, чтобы сохранить medium, затем перейдите на вкладку *Permissions*.

На вкладке Permissions есть обязательное поле *Role*. Роль определяет, какие элементы веб-интерфейса пользователь может просматривать и какие действия ему разрешено выполнять. Нажмите *Select* и выберите одну из ролей из списка. Например, выберите *Admin role*, чтобы разрешить доступ ко всем разделам веб-интерфейса Zabbix, кроме *Administration*. Позже вы сможете изменить разрешения или создать дополнительные роли пользователей. После выбора роли разрешения появятся на той же вкладке:

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

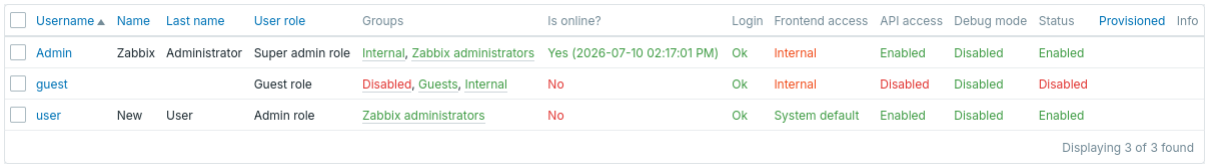
Enabled | Manage own devices

Access to actions

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

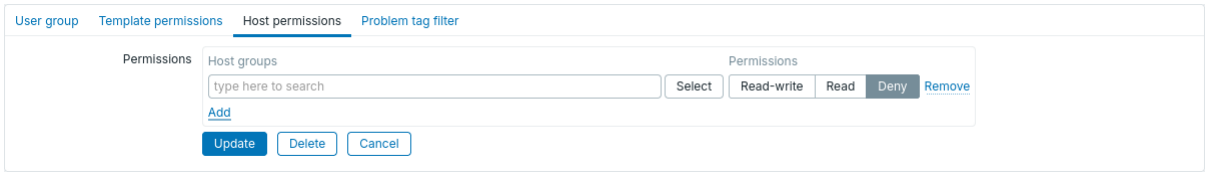
Add | Cancel

Нажмите *Add* в форме свойств пользователя, чтобы сохранить пользователя. Новый пользователь появится в списке пользователей.

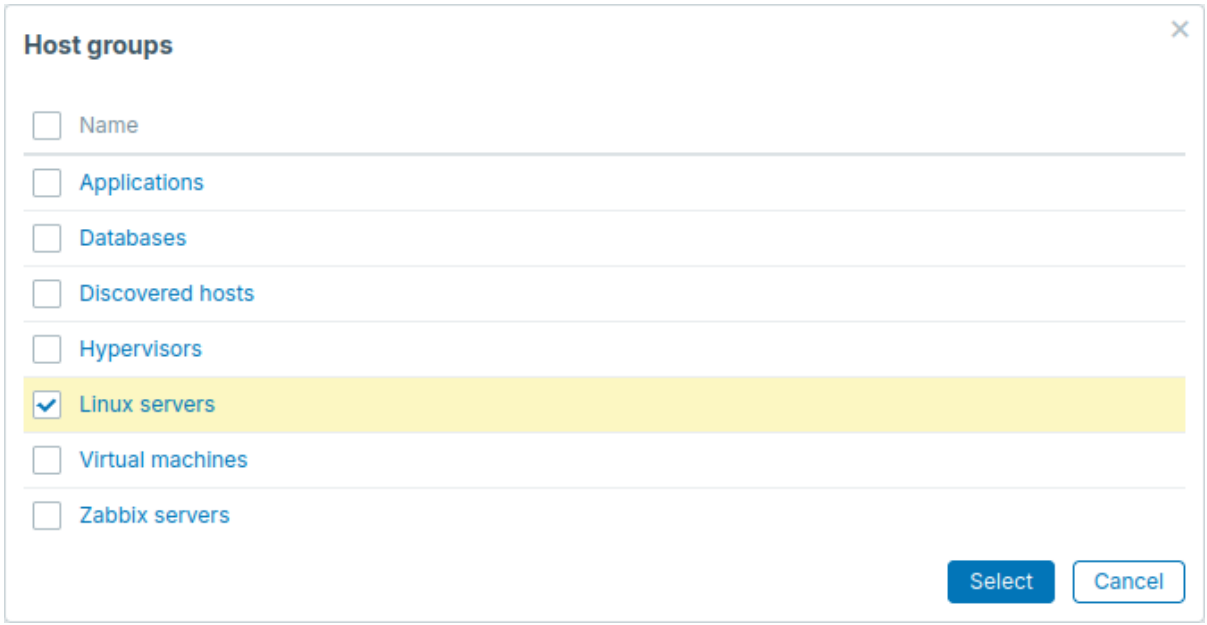


Добавление прав доступа

По умолчанию новый пользователь не имеет прав доступа к узлам сети и шаблонам. Чтобы предоставить пользователю права, нажмите на группу пользователя в столбце *Groups* (в данном случае - *Zabbix administrators*). В форме свойств *User groups* перейдите на вкладку *Host permissions*, чтобы назначить права для групп узлов сети. Нажмите Add в поле выбора группы узлов сети, чтобы отобразить его:



Затем нажмите *Select* рядом с полем, чтобы увидеть список групп узлов сети. Для этого пользователя требуется доступ только для чтения к группе *Linux servers*, поэтому отметьте соответствующий флажок в списке и нажмите *Select*, чтобы подтвердить выбор.



Нажмите кнопку *Read*, чтобы задать уровень прав доступа, а затем *Update*, чтобы сохранить изменения, внесенные в конфигурацию группы пользователей.

Чтобы предоставить права доступа к шаблонам, нужно перейти на вкладку *Template permissions* и указать группы шаблонов. Шаги аналогичны назначению прав для групп узлов сети. Обзор шаблонов доступен в разделе New template этого Quickstart.

Attention:
В Zabbix права доступа к узлам сети и шаблонам назначаются группам пользователей, а не отдельным пользователям.

Готово! Теперь вы можете попробовать войти в систему, используя учетные данные нового пользователя.

Новый узел сети

Обзор

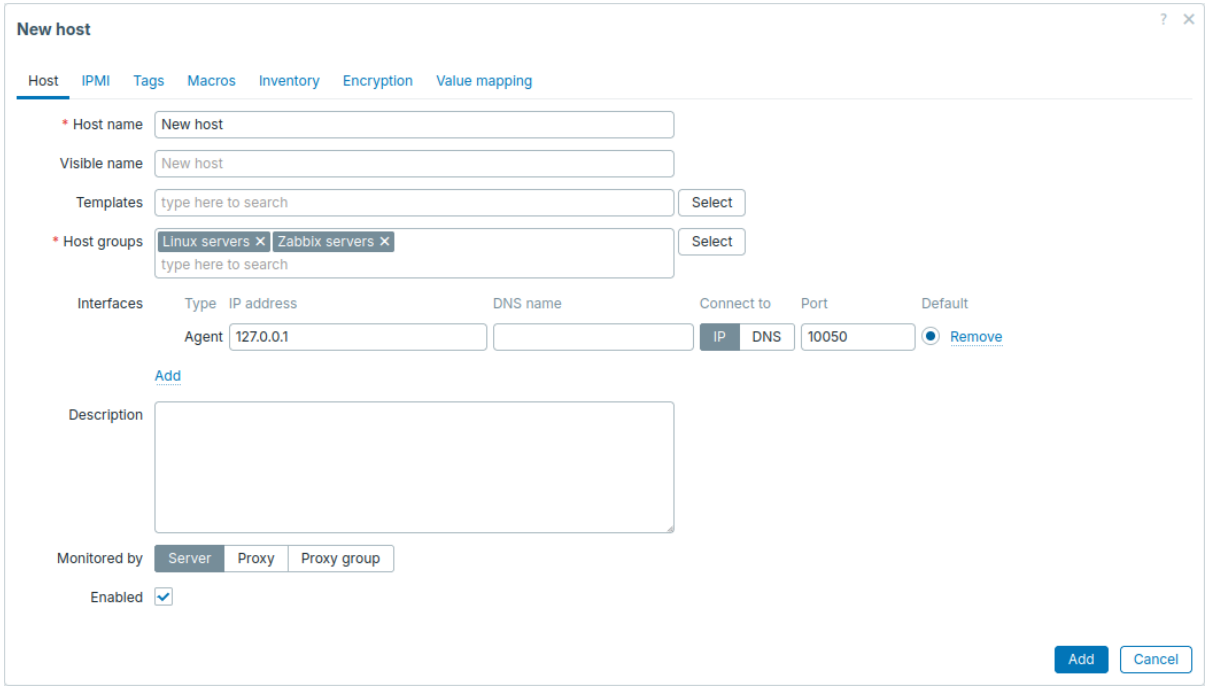
В этом разделе вы узнаете, как настроить новый узел сети.

Узел сети в Zabbix — это сетевой объект (физический, виртуальный), который вы хотите мониторить. Определение того, что может быть "узлом сети" в Zabbix, довольно гибкое. Это может быть физический сервер, сетевой коммутатор, виртуальная машина или какое-либо приложение.

Добавление узла сети

Информация о настроенных узлах сети в Zabbix доступна в разделе меню *Сбор данных* > *Узлы сети*, а также в разделе меню *Мониторинг* > *Узлы сети*. В системе уже есть один предварительно настроенный узел сети с именем «Zabbix server», но мы рассмотрим добавление еще одного.

Чтобы добавить новый узел сети, нажмите *Создать узел сети*. Откроется форма настройки узла сети.



Все обязательные поля отмечены красной звездочкой.

Минимально необходимо указать следующее:

Имя узла сети

- Введите имя узла сети. Разрешены буквы и цифры, пробелы, точки, дефисы и символы подчеркивания.

Группы узлов сети

- Выберите одну или несколько существующих групп, нажав кнопку *Выбрать*, или введите имя несуществующей группы, чтобы создать новую группу.

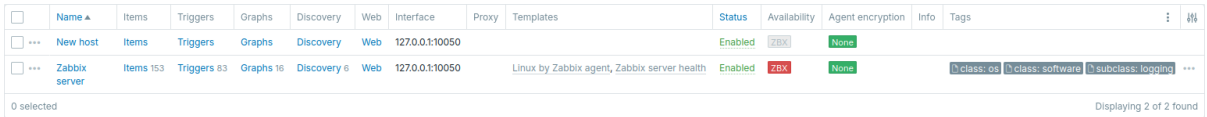
Note:
Все права доступа назначаются группам узлов сети, а не отдельным узлам сети. Поэтому узел сети должен принадлежать как минимум одной группе.

Интерфейсы: IP-адрес

- Хотя технически это поле не является обязательным, интерфейс узла сети необходим для сбора определенных метрик. Чтобы использовать пассивные проверки с помощью агента Zabbix, укажите в этом поле IP-адрес или DNS-имя агента. Также необходимо указать IP-адрес или DNS-имя сервера Zabbix в директиве `Server` файла конфигурации агента. Если агент Zabbix и сервер Zabbix установлены на одном компьютере, необходимо указать один и тот же IP-адрес или DNS-имя в обоих местах.

Другие параметры можно оставить без изменений, используя их значения по умолчанию.

После этого нажмите *Добавить*. Новый узел сети должен появиться в списке узлов сети.



В столбце *Доступность* отображаются индикаторы доступности узла сети для каждого интерфейса. Если определен интерфейс агента Zabbix, для определения доступности узла сети можно использовать значок доступности агента (с меткой *ZBX*):

- ZBX - состояние узла сети еще не определено; проверка метрики еще не выполнялась.
- ZBX - узел сети доступен, проверка метрики выполнена успешно.
- ZBX - узел сети недоступен, проверка метрики завершилась ошибкой (наведите указатель мыши на значок, чтобы просмотреть сообщение об ошибке). Возможно, возникла ошибка связи, причиной которой могут быть неверные учетные данные интерфейса. Убедитесь, что сервер Zabbix запущен, а затем попробуйте обновить страницу позже.

Новый элемент данных

Обзор

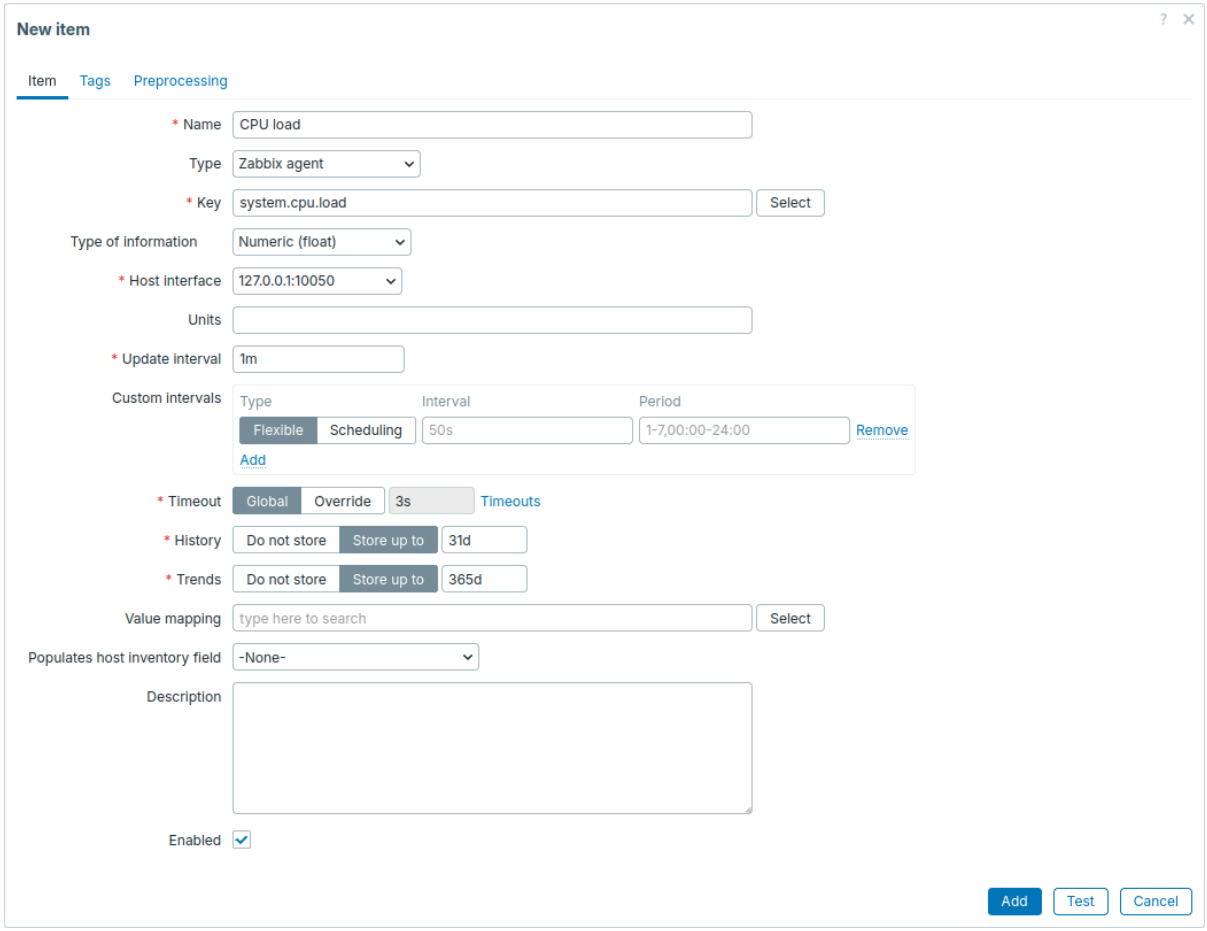
В этом разделе вы узнаете, как настроить элемент данных.

Элементы данных являются основой сбора данных в Zabbix. Без элементов данных нет и данных — поскольку только элемент данных определяет отдельную метрику или то, какие данные следует собирать с узла сети.

Добавление элемента данных

Все элементы данных сгруппированы вокруг узлов сети. Именно поэтому для настройки примера элемента данных мы переходим в *Сбор данных* > *Узлы сети* и находим созданный нами "New host".

Нажмите ссылку *Элементы данных* в строке "New host", а затем нажмите *Создать элемент данных*. Откроется форма настройки элемента данных.



Все обязательные поля ввода отмечены красной звездочкой.

Для нашего примера элемента данных необходимо ввести следующую основную информацию:

Имя

- Введите значение *CPU load*. Это имя элемента данных, которое будет отображаться в списках и в других местах.

Ключ

- Введите вручную значение *system.cpu.load*. Это техническое имя элемента данных, которое определяет тип собираемой информации. Указанный ключ — это один из предопределенных ключей, поставляемых вместе с агентом Zabbix.

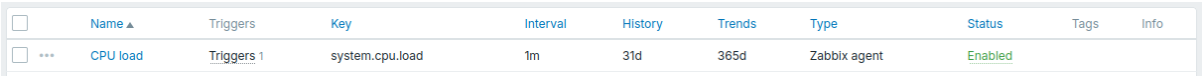
Тип информации

- Этот атрибут определяет формат ожидаемых данных. Для ключа *system.cpu.load* это поле будет автоматически установлено в значение *Numeric (float)*.

Note:
Также рекомендуется уменьшить количество дней хранения истории элемента данных до 7 или 14. Это хорошая практика, позволяющая снизить нагрузку на базу данных, которой не придется хранить большое количество исторических значений.

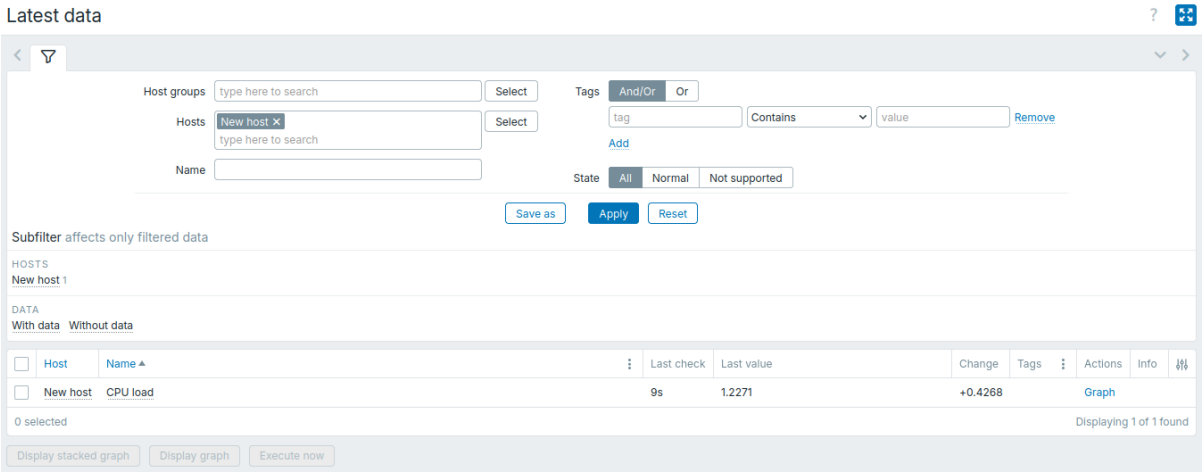
Другие параметры можно оставить без изменений, используя значения по умолчанию.

После завершения нажмите *Добавить*. Новый элемент данных должен появиться в списке элементов данных.



Просмотр данных

После того как элемент данных определен, вам может быть интересно, действительно ли он собирает данные. Для этого перейдите в *Monitoring* > *Latest data*, выберите 'New host' в фильтре и нажмите *Apply*.



При этом может потребоваться до 60 секунд, прежде чем поступят первые данные. По умолчанию именно с такой периодичностью сервер считывает изменения конфигурации и обнаруживает новые элементы данных для выполнения.

Если в столбце 'Change' нет значения, возможно, пока получено только одно значение. Подождите 30 секунд, пока не поступит еще одно значение.

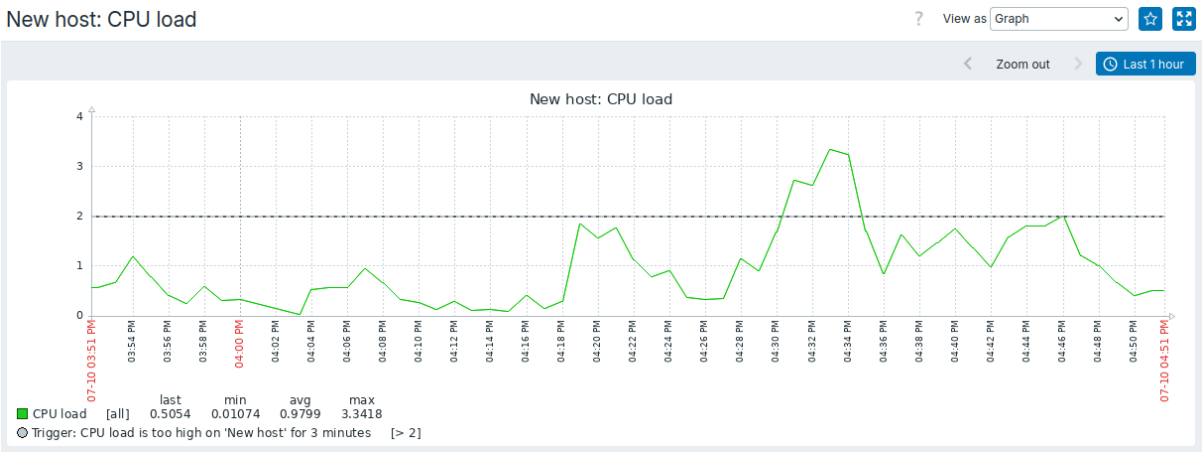
Если вы не видите информацию об элементе данных, как на снимке экрана, убедитесь, что:

- вы заполнили поля элемента данных 'Key' и 'Type of information' точно так же, как на снимке экрана;
- и агент, и сервер запущены;
- статус узла сети — 'Monitored', а значок доступности зеленый;
- выбранный в фильтре узел сети указан правильно;
- элемент данных включен.

Графики

После того как элемент данных поработает некоторое время, возможно, вы захотите увидеть визуальное представление данных. Простые графики доступны для любого отслеживаемого числового элемента данных без дополнительной настройки. Эти графики создаются во время выполнения.

Чтобы просмотреть график, перейдите в раздел *Мониторинг* > *Последние данные* и нажмите ссылку *График* рядом с элементом данных.



Новый триггер

Обзор

В этом разделе вы узнаете, как настроить триггер.

Элементы данных только собирают данные. Чтобы автоматически оценивать поступающие данные, нам нужно определить триггеры. Триггер содержит выражение, которое задает порог допустимого уровня для данных.

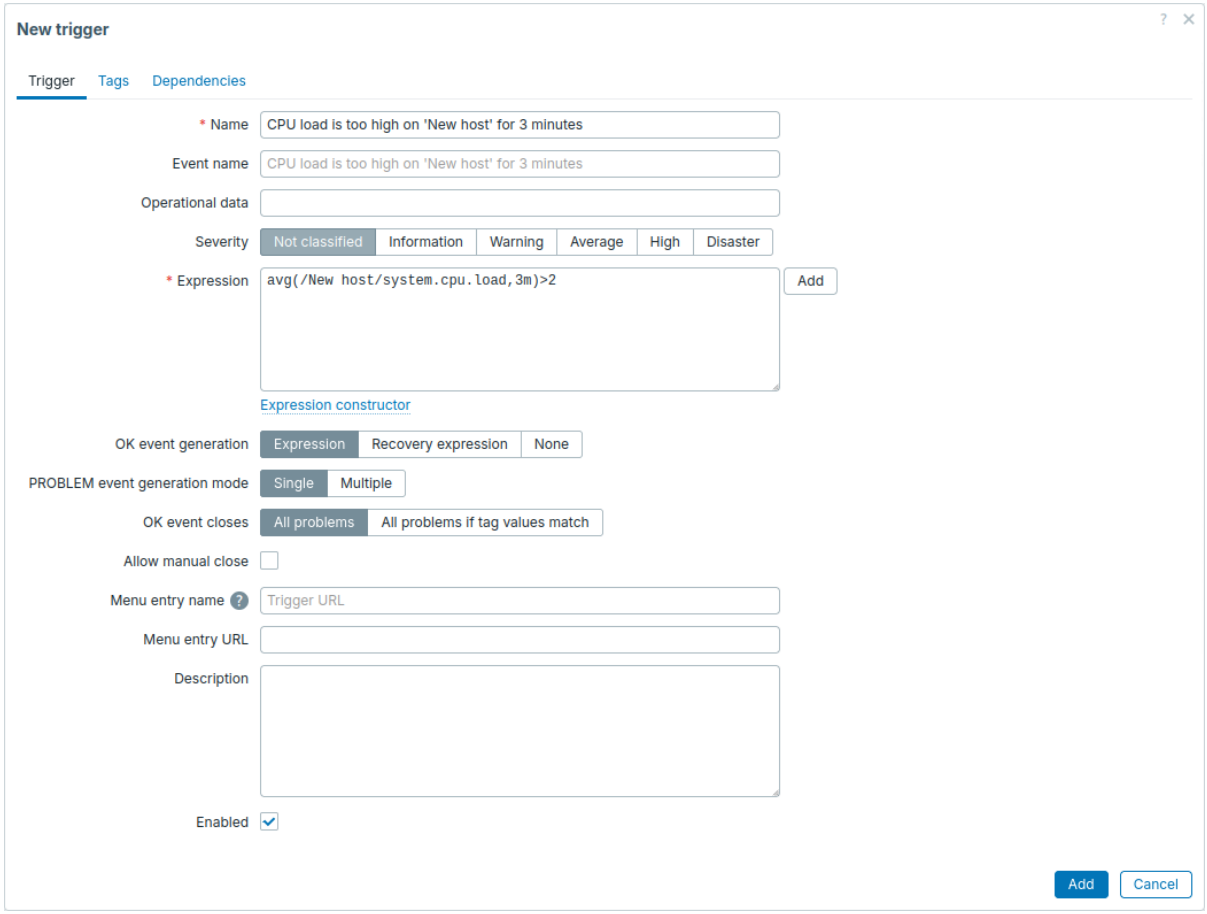
Если этот уровень будет превышен поступающими данными, триггер ”сработает” или перейдет в состояние ’Problem’ — это означает, что произошло что-то, что может потребовать внимания. Если уровень снова станет допустимым, триггер возвращается в состояние ’Ok’.

Добавление триггера

Чтобы настроить триггер для нашего элемента данных:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Нажмите ссылку *Триггеры* в строке ’New host’.
3. Нажмите *Создать триггер*.

Откроется форма настройки триггера.



Для нашего триггера здесь нужно указать следующую основную информацию:

Имя

- Введите *CPU load is too high on 'New host' for 3 minutes* в качестве значения. Это будет имя триггера, отображаемое в списках и в других местах.

Выражение

- Введите: `avg(/New host/system.cpu.load,3m)>2`

Это выражение триггера. Убедитесь, что выражение введено правильно, вплоть до последнего символа. Ключ элемента данных здесь (`system.cpu.load`) используется для ссылки на элемент данных. По сути, это конкретное выражение означает, что порог проблемы превышен, когда среднее значение загрузки CPU за 3 минуты больше 2. Подробнее о синтаксисе выражений триггеров.

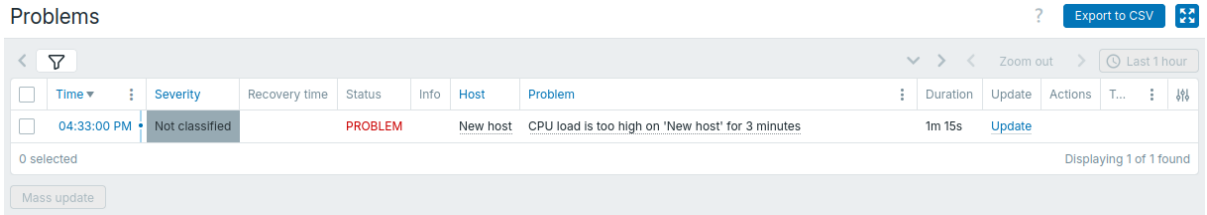
Другие параметры можно оставить без изменений, используя значения по умолчанию.

После завершения нажмите *Добавить*. Новый триггер должен появиться в списке триггеров.

Отображение состояния триггера

После определения триггера вам может быть интересно посмотреть его состояние.

Если загрузка CPU превысила пороговый уровень, заданный в триггере, проблема будет отображаться в *Мониторинг* > *Проблемы*.



Мигание в столбце состояния указывает на недавнее изменение состояния триггера, произошедшее в течение последних 30 минут.

Получение уведомления о проблеме

Обзор

В этом разделе вы узнаете, как настроить оповещения в виде уведомлений в Zabbix.

Поскольку элементы данных собирают данные, а триггеры предназначены для срабатывания при возникновении проблемных ситуаций, полезно также иметь механизм оповещения, который будет уведомлять о важных событиях даже тогда, когда веб-интерфейс Zabbix не проверяется активно.

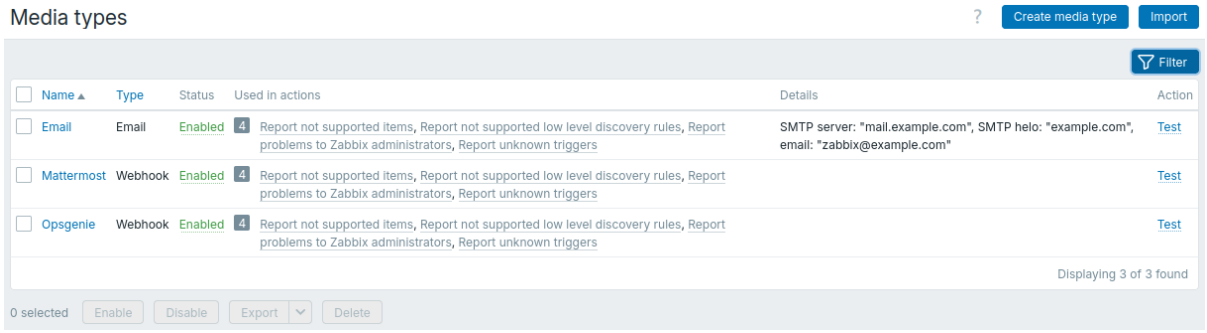
Именно это и делают уведомления.

Поскольку электронная почта является наиболее популярным способом доставки уведомлений о проблемах, мы рассмотрим, как настроить уведомление по электронной почте.

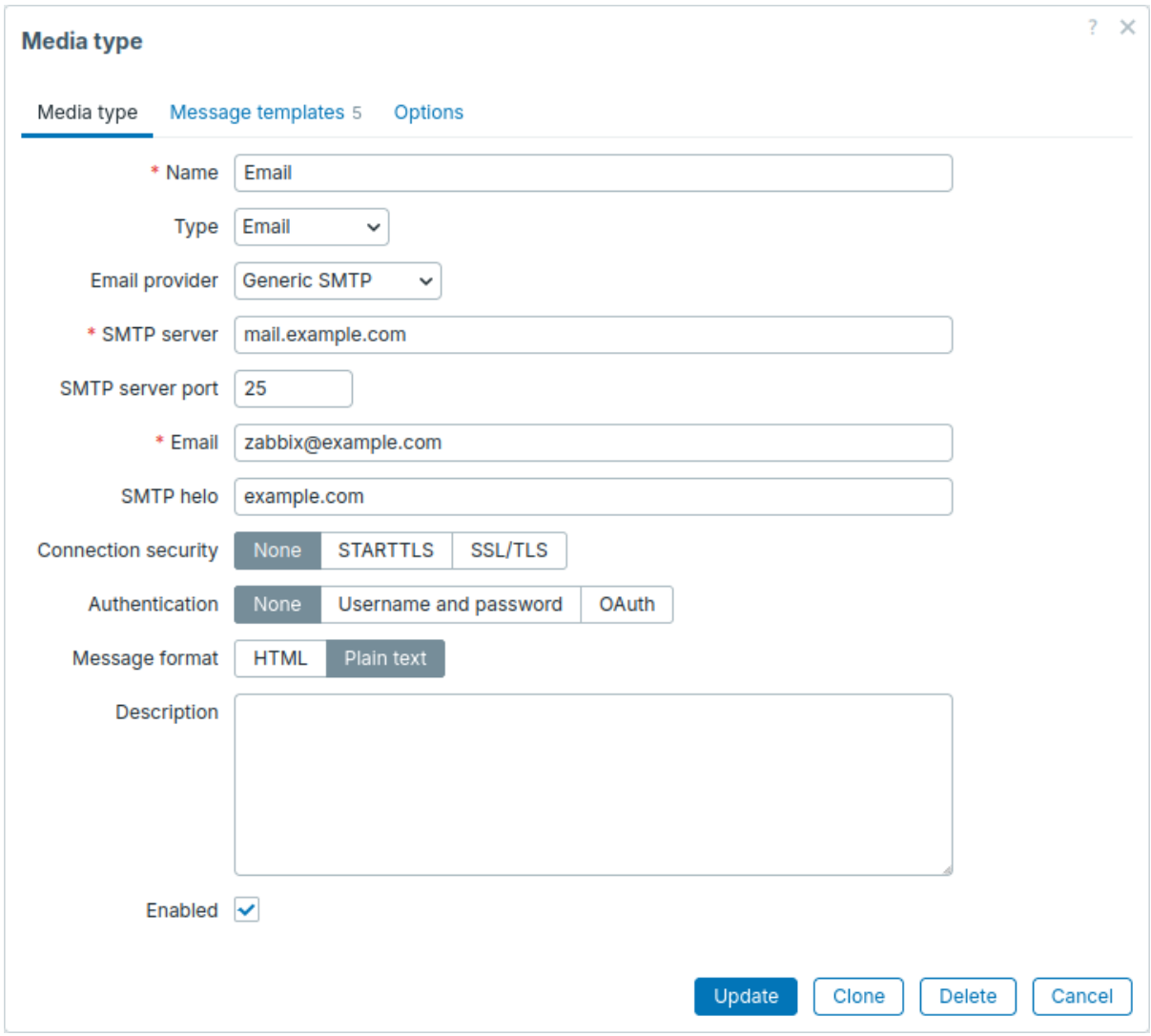
Настройки электронной почты

Изначально в Zabbix уже есть несколько предопределенных способов доставки уведомлений. Email — один из них.

Чтобы настроить параметры электронной почты, перейдите в *Оповещения* > *Типы медиа* и нажмите *Email* в списке предопределенных типов медиа.



Откроется форма настройки параметров электронной почты.



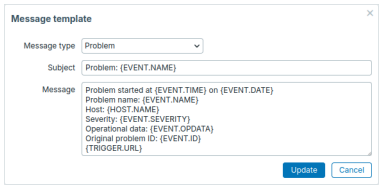
Все обязательные поля ввода отмечены красной звездочкой.

На вкладке *Тип медиа* задайте значения *SMTP server*, *SMTP helo* и *Email* в соответствии с вашей средой.

Note:
Значение в поле *Email* будет использоваться как адрес 'From' для уведомлений, отправляемых из Zabbix.

Далее необходимо определить содержимое сообщения о проблеме. Содержимое задается с помощью шаблона сообщения, настраиваемого на вкладке *Message templates*.

Нажмите *Add*, чтобы создать шаблон сообщения, и выберите *Problem* в качестве типа сообщения.

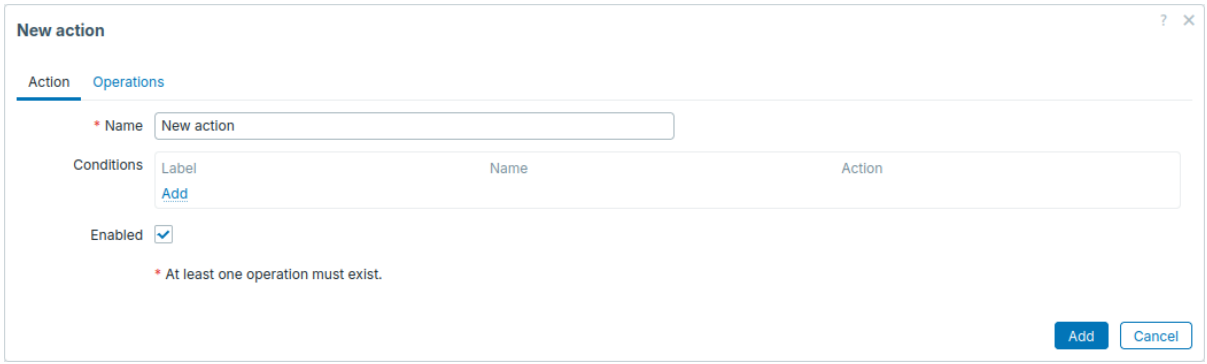


Когда все будет готово, нажмите *Add* и сохраните форму.

Теперь вы настроили *Email* как рабочий тип медиа. Тип медиа также должен быть привязан к пользователям путем указания конкретных адресов доставки (как мы делали при создании нового пользователя); в противном случае он не будет использоваться.

Новое действие

Доставка уведомлений — одна из задач, которые выполняют действия в Zabbix. Поэтому, чтобы настроить уведомление, перейдите в *Оповещения* > *Действия* > *Действия триггеров* и нажмите *Создать действие*.

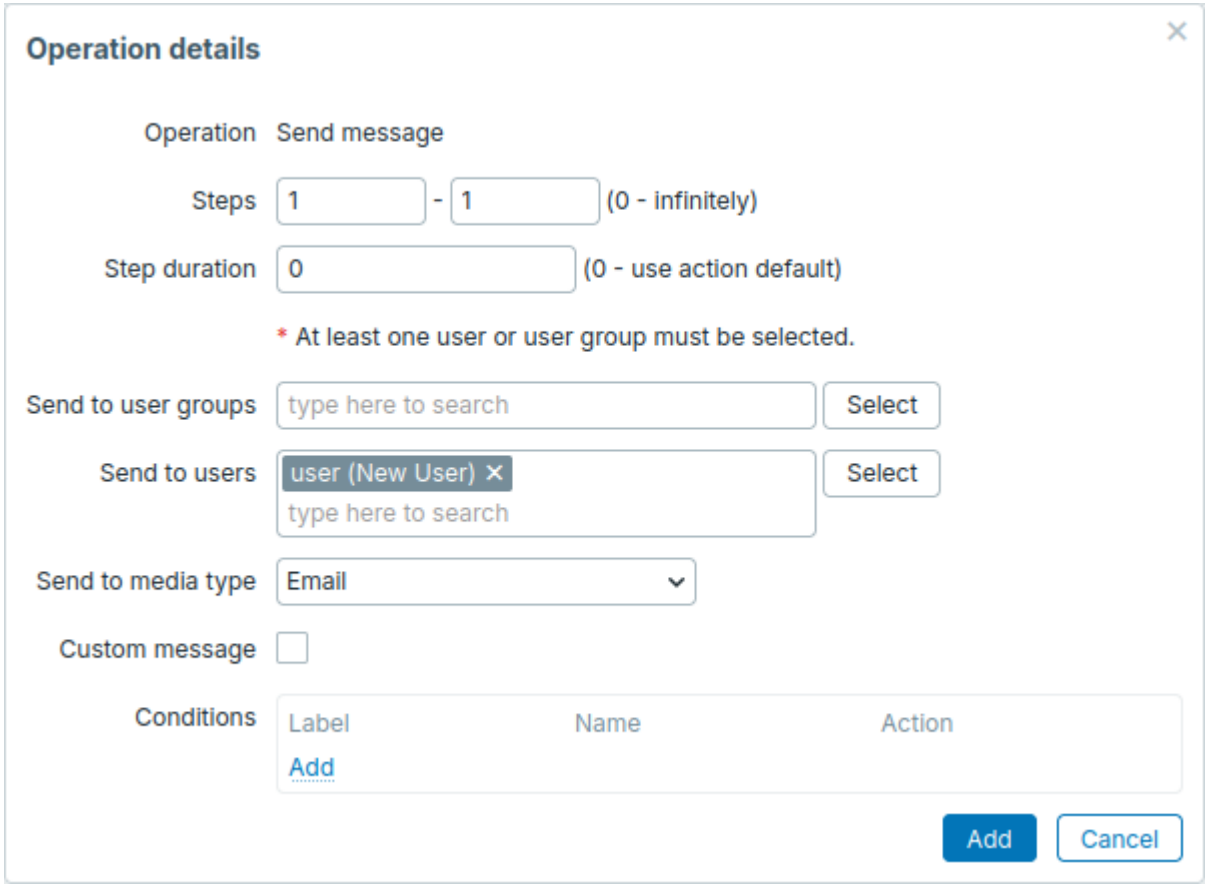


Все обязательные поля ввода отмечены красной звездочкой.

В этой форме укажите имя действия.

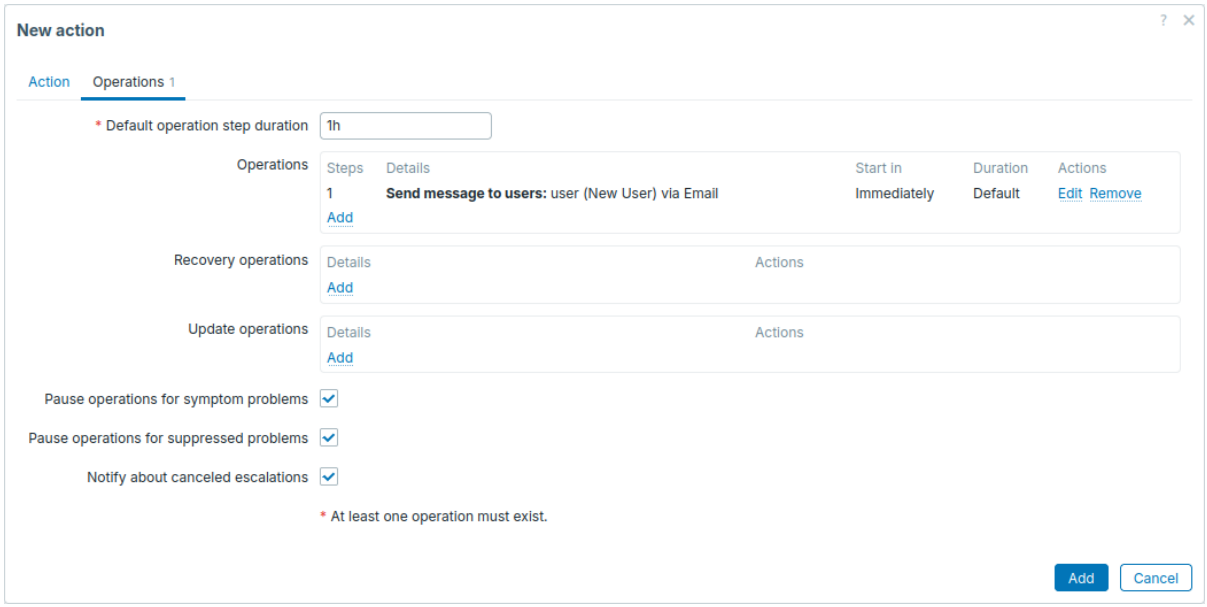
В самом простом случае, если не добавлять никаких более конкретных условий, действие будет выполнено при любом изменении триггера с 'Ok' на 'Problem'.

Нам также нужно определить, что именно должно делать действие — это задается на вкладке *Операции*. Нажмите *Добавить* в блоке *Операции*, чтобы открыть новую форму операции.



Все обязательные поля ввода отмечены красной звездочкой.

Здесь нажмите *Выбрать* в блоке *Отправить пользователям* и выберите пользователя (‘user’), которого мы определили. В качестве значения *Отправить через тип медиа* выберите ”Email”. После этого нажмите *Добавить*, и операция должна быть добавлена:



На этом настройка простого действия завершена, поэтому нажмите *Добавить* в форме действия.

Получение уведомления

Теперь, когда доставка уведомлений настроена, было бы полезно фактически получить одно из них. Для этого мы можем намеренно увеличить нагрузку на наш узел сети, чтобы наш триггер сработал и мы получили уведомление о проблеме.

Откройте консоль на вашем узле сети и выполните:

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

Вы можете запустить один или несколько из этих процессов.

Теперь перейдите в *Monitoring* > *Latest data* и посмотрите, как увеличились значения 'CPU Load'. Помните, чтобы наш триггер сработал, значение 'CPU Load' должно быть выше '2' в течение 3 минут подряд. Как только это произойдет:

- в *Monitoring* > *Problems* вы должны увидеть триггер с мигающим состоянием `Problem`;
- вы должны получить уведомление о проблеме по электронной почте.

Attention:
Если уведомления не работают:

- еще раз проверьте, что и настройки электронной почты, и действие настроены правильно
- убедитесь, что созданный вами пользователь имеет как минимум права на чтение на узле сети, который сгенерировал событие, как указано на шаге *Adding user*. Пользователь, будучи частью группы пользователей 'Zabbix administrators', должен иметь как минимум доступ на чтение к группе узлов сети 'Linux servers', к которой принадлежит наш узел сети.
- Кроме того, вы можете посмотреть журнал действий, перейдя в *Reports* > *Action log*.

Новый шаблон

Обзор

В этом разделе вы узнаете, как настроить шаблон.

Ранее мы узнали, как настроить элемент данных и триггер, а также как получать уведомления о проблемах для узла сети.

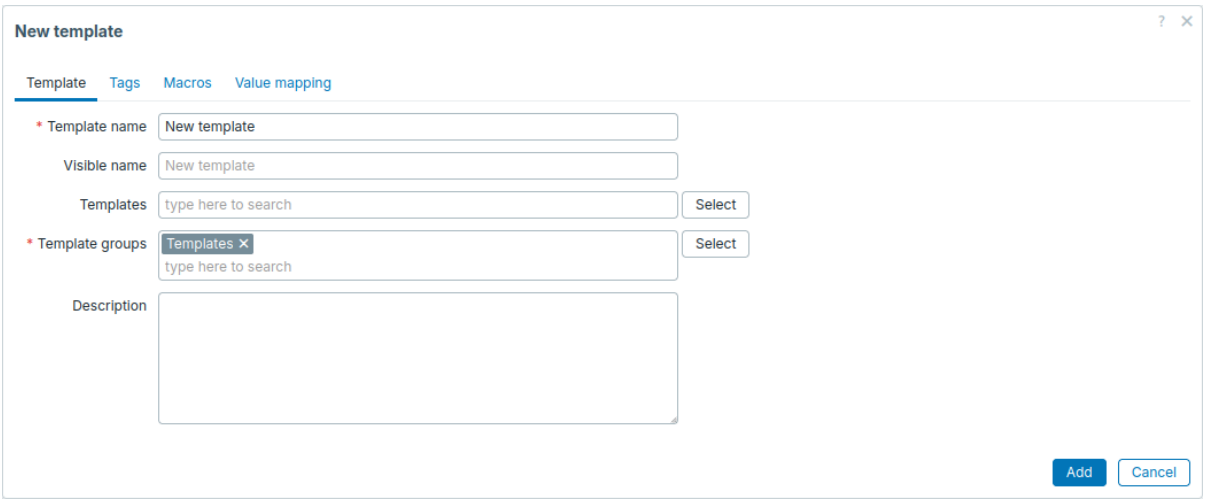
Хотя все эти шаги сами по себе обеспечивают широкие возможности, их может показаться слишком много, если их потребуется выполнить, например, для тысячи узлов сети. Здесь пригодилась бы автоматизация.

Именно здесь на помощь приходят шаблоны. Шаблоны позволяют группировать полезные элементы данных, триггеры и другие сущности, чтобы многократно использовать их, применяя к узлам сети за один шаг.

Когда шаблон связывается с узлом сети, узел сети наследует все сущности шаблона. Таким образом, заранее подготовленный набор проверок можно очень быстро применить.

Добавление шаблона

Чтобы начать работу с шаблонами, сначала нужно создать шаблон. Для этого в *Сбор данных* > *Шаблоны* нажмите *Создать шаблон*. Откроется форма настройки шаблона.



Все обязательные поля ввода отмечены красной звездочкой.

Необходимые параметры, которые нужно указать здесь:

Имя шаблона

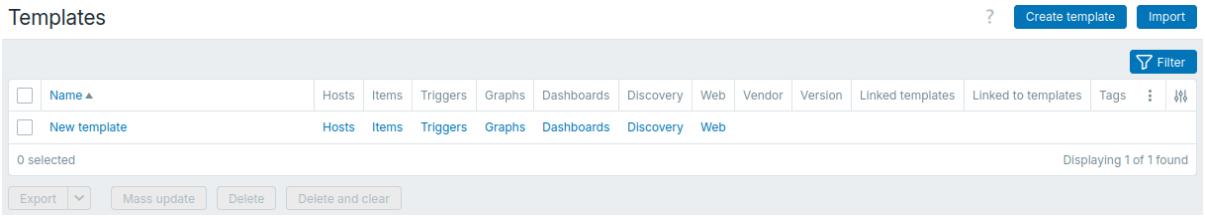
- Введите имя шаблона. Допускаются буквенно-цифровые символы, пробелы и символы подчеркивания.

Группы шаблонов

- Выберите одну или несколько групп, нажав кнопку *Выбрать*. Шаблон должен принадлежать к группе.

Note:
Права доступа к группам шаблонов назначаются в настройках группы пользователей на вкладке **Права на шаблоны** так же, как и права на узлы сети. Все права доступа назначаются группам, а не отдельным шаблонам, поэтому включение шаблона хотя бы в одну группу обязательно.

После завершения нажмите *Добавить*. Новый шаблон должен отображаться в списке шаблонов. Также можно использовать фильтр, чтобы найти свой шаблон.

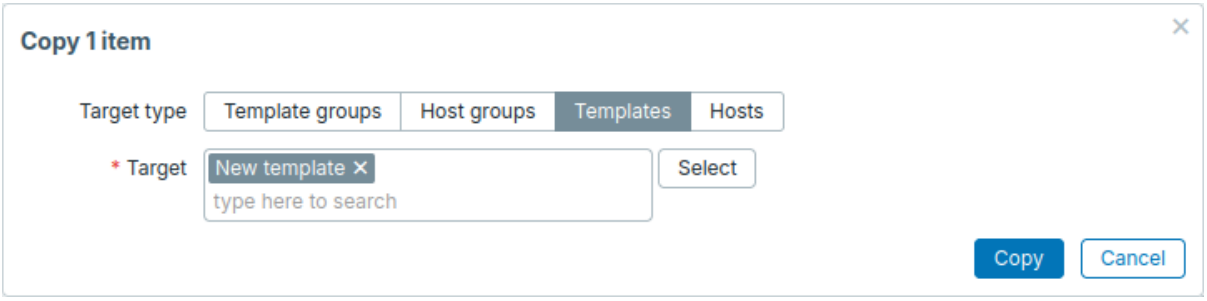


Как видите, шаблон уже есть, но он пока пуст — в нем нет ни элементов данных, ни триггеров, ни других сущностей.

Добавление элемента данных в шаблон

Чтобы добавить элемент данных в шаблон:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Найдите в списке «Новый узел сети» и нажмите гиперссылку *Элементы данных*.
3. Установите флажок элемента данных «Загрузка ЦП» в списке.
4. Нажмите *Копировать* под списком.
5. Перейдите на вкладку *Шаблоны*.
6. Выберите шаблон, в который нужно скопировать элемент данных.



Все обязательные поля ввода отмечены красной звездочкой. 7. Нажмите *Копировать*.

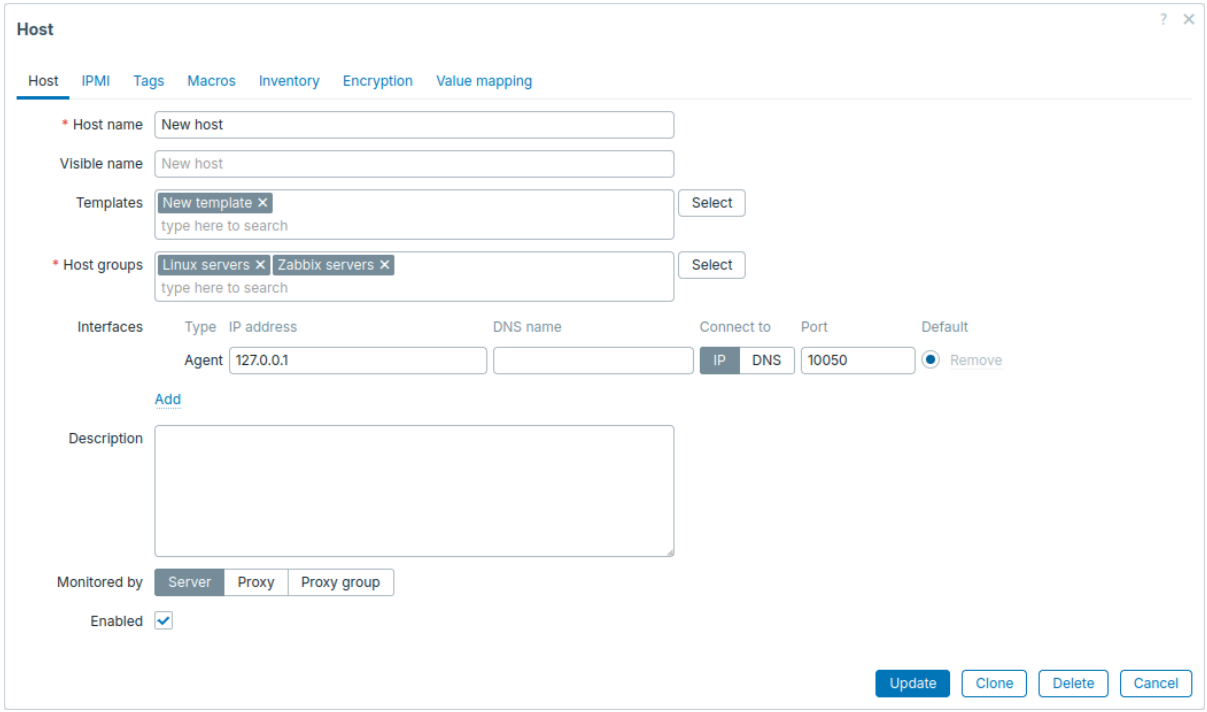
Если теперь перейти в раздел *Сбор данных* > *Шаблоны*, в «Новом шаблоне» должен появиться один новый элемент данных.

Пока мы остановимся только на одном элементе данных, но аналогичным образом можно добавлять в шаблон любые другие элементы данных, триггеры или другие сущности, пока не будет сформирован достаточно полный набор сущностей для определенной цели (мониторинг ОС, мониторинг отдельного приложения).

Связывание шаблона с узлом сети

После подготовки шаблона остается только добавить его к узлу сети. Для этого:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Нажмите «Новый узел сети», чтобы открыть форму его настройки, и найдите поле *Шаблоны*.
3. Начните вводить *Новый шаблон* в поле *Шаблоны*. Имя созданного шаблона должно появиться в раскрывающемся списке.
4. Прокрутите список вниз и выберите шаблон. Убедитесь, что он появился в поле *Шаблоны*.



5. Нажмите *Обновить* в форме, чтобы сохранить изменения.

Теперь шаблон добавлен к узлу сети вместе со всеми содержащимися в нем сущностями.

Таким образом, его можно применить и к любому другому узлу сети. Все изменения элементов данных, триггеров и других сущностей на уровне шаблона будут распространяться на узлы сети, с которыми связан шаблон.

Привязка предопределенных шаблонов к узлам сети

Как вы могли заметить, Zabbix поставляется с набором предопределенных шаблонов для различных ОС, устройств и приложений. Чтобы очень быстро начать мониторинг, вы можете привязать к узлу сети подходящий шаблон, но учтите, что эти шаблоны нужно донастроить под вашу среду. Некоторые проверки могут быть не нужны, а интервалы опроса могут быть слишком частыми.

Доступна дополнительная информация о шаблонах.

Мониторинг веб-сервера Apache

Чтобы начать мониторинг веб-сервера Apache, выполните следующие действия:

1. Убедитесь, что доступ к статистике Apache открыт.
2. Примените шаблон мониторинга Apache.
3. Просмотрите последние данные.

Чтобы использовать это руководство, у вас должны быть установлены сервер Zabbix и веб-интерфейс installed. Вы должны войти в Zabbix как пользователь *Admin* по умолчанию.

Доступ к статистике Apache

Цель этого шага — убедиться, что статистика веб-сервера Apache может быть прочитана Zabbix.

В Apache актуальная информация о производительности и активности сервера предоставляется встроенным модулем `mod_status`. При включении он создает доступную через веб страницу (обычно по адресу `/server-status`), на которой отображаются данные о производительности и активности.

Чтобы проверить, включен ли модуль `mod_status` для веб-сервера Apache, выполните:

В Debian/Ubuntu	В системах на базе RHEL
`apache2ctl -M | grep status`	`httpd -M | grep status`

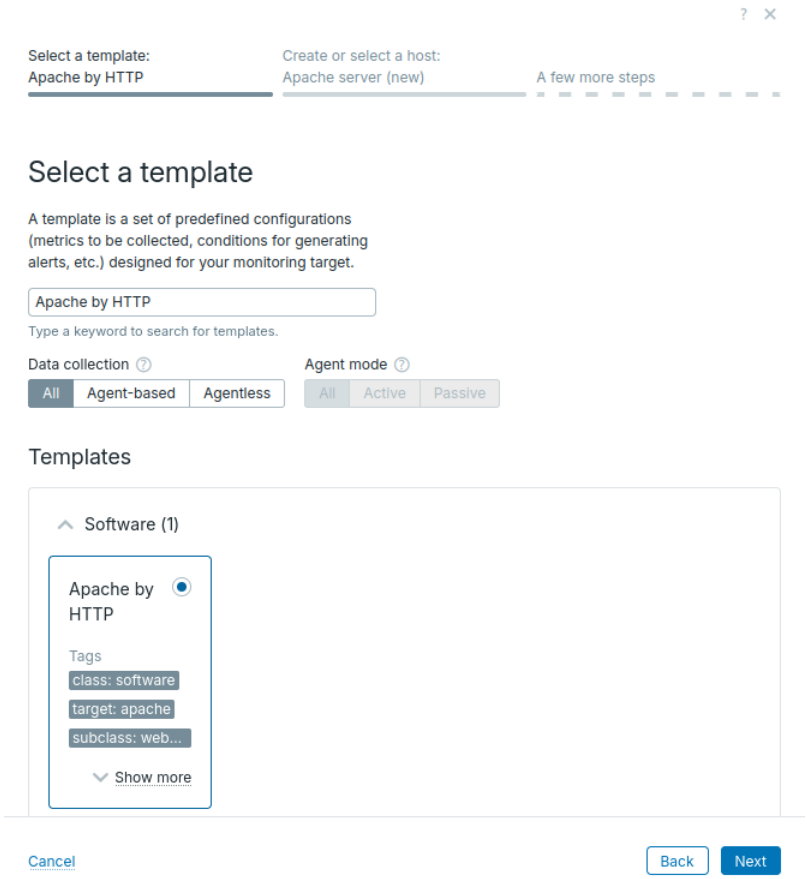
Если в выводе вы видите "`status_module (shared)`", значит модуль status уже включен. Если нет, то вам нужно включить модуль status.

Применение шаблона мониторинга Apache

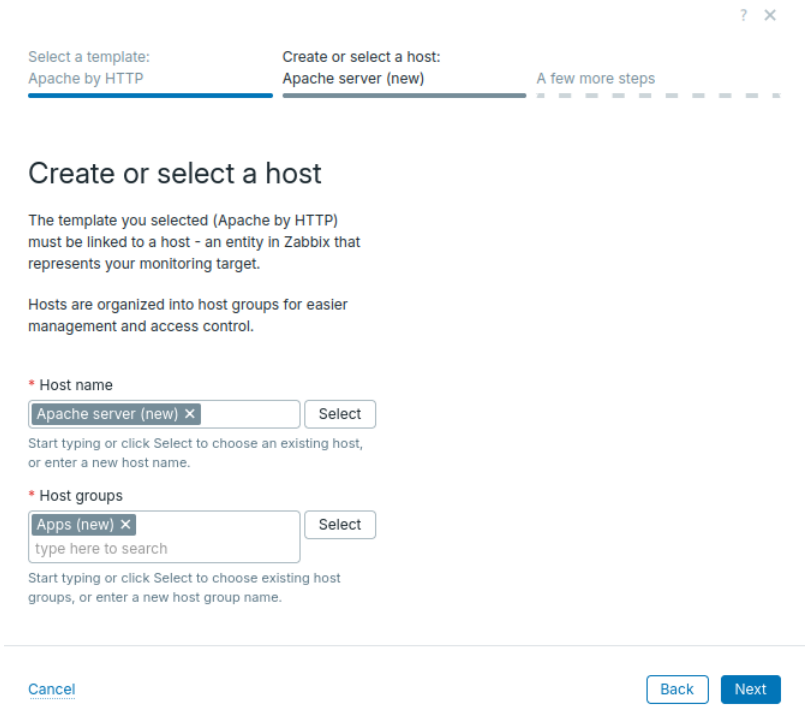
В Zabbix этот шаблон готов к использованию. Он содержит предварительно настроенные элементы, готовые для сбора данных, визуализации и анализа.

Чтобы применить шаблон:

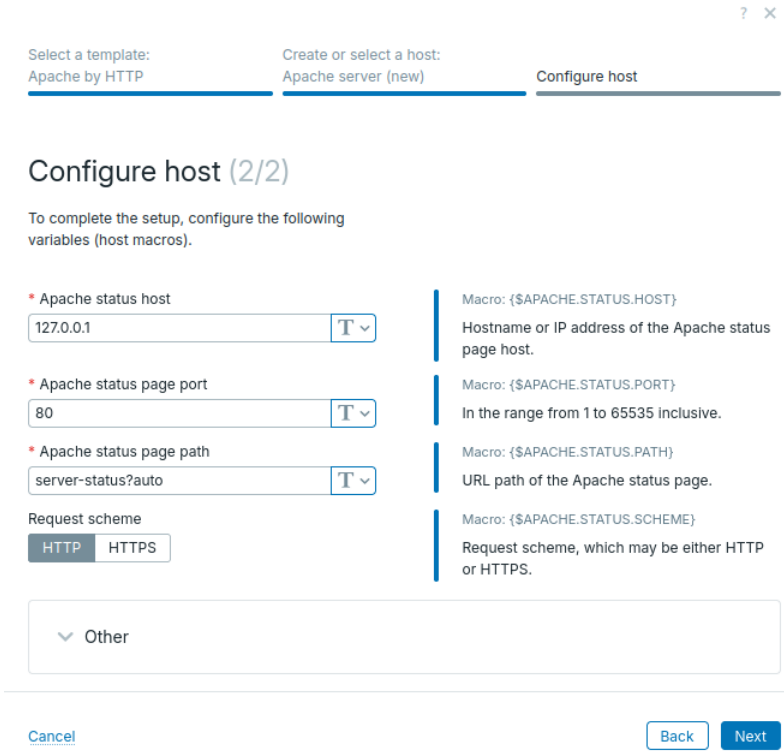
1. Запустите Мастер узлов сети в разделе *Сбор данных* > *Узлы сети*.
2. Выберите шаблон *Apache by HTTP*.



3. Создайте узел сети Apache и добавьте для него группу. Этот виртуальный узел сети будет содержать данные, связанные с Apache.



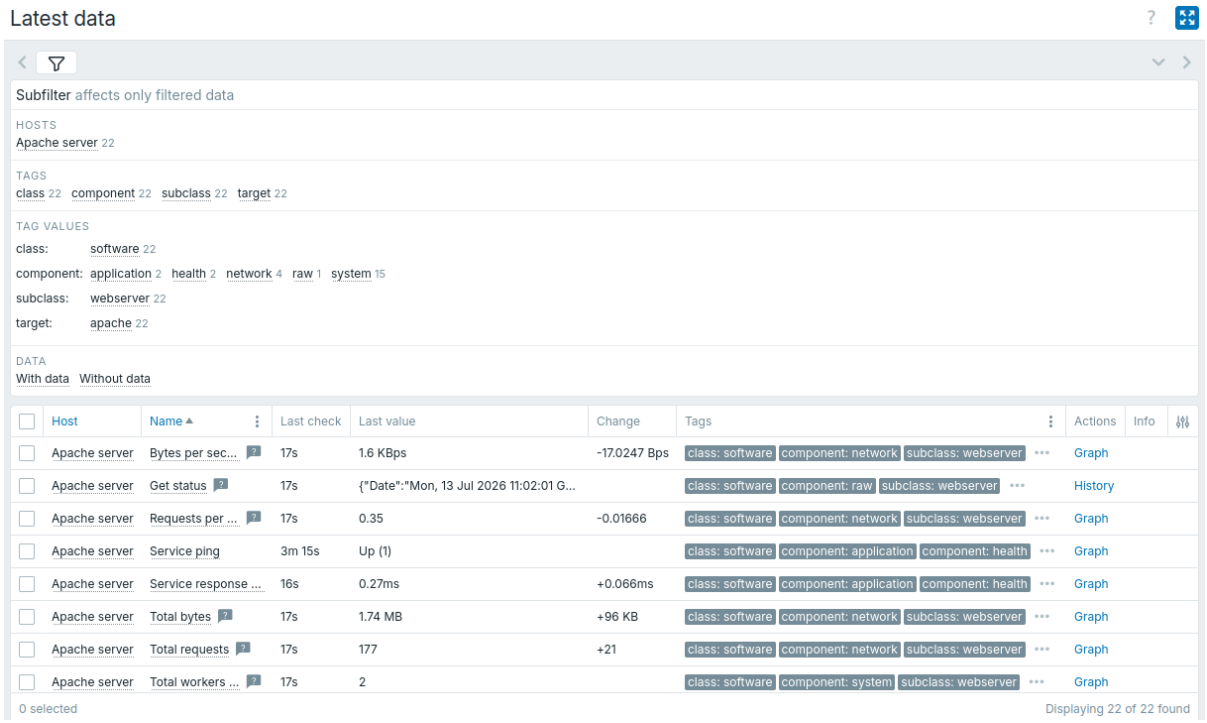
4. Настройте узел сети. Важно указать адрес веб-сервера Apache (127.0.0.1 для локальной машины, IP-адрес для удаленной машины).



Остальные параметры можно оставить по умолчанию.

5. Чтобы завершить работу Мастера узлов сети, нажмите *Create* и *Finish*.

Просмотр последних данных



Поздравляем, мониторинг веб-сервера Apache запущен!

Zabbix предоставляет возможности визуализации и отправляет оповещения при возникновении любых проблем. Чтобы настроить оповещения по электронной почте, см. Получение уведомлений о проблемах.

Мониторинг Linux-сервера

Чтобы начать мониторинг Linux-сервера, выполните следующие действия:

1. Примените шаблон мониторинга Linux.
2. Установите агент Zabbix на сервер.

3. Просмотрите последние данные.

Чтобы начать мониторинг удаленного Linux-сервера, вам нужно знать его IP-адрес и IP-адрес сервера Zabbix.

Для мониторинга Linux-сервера будет использоваться агент Zabbix. Инструкции по установке агента будут приведены в процессе.

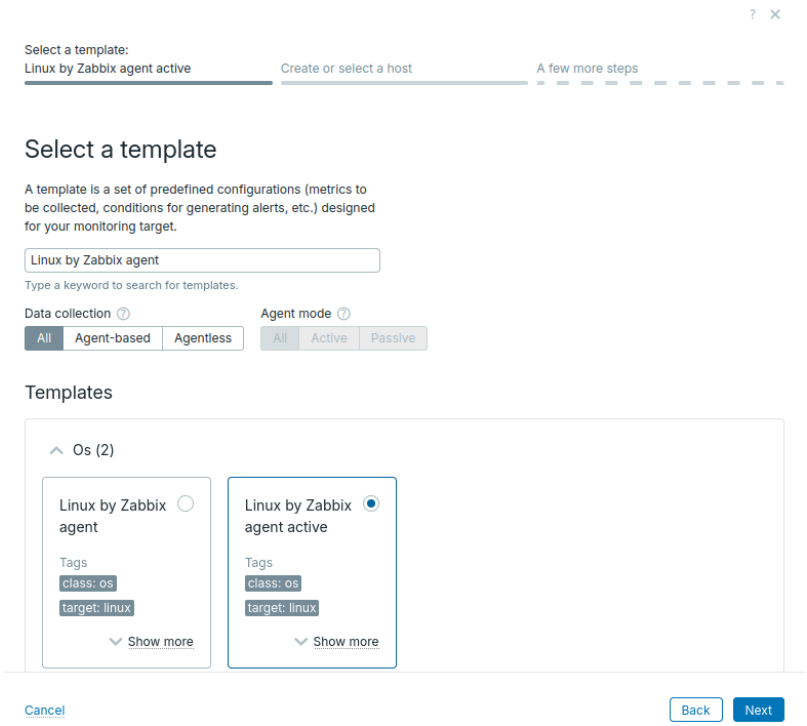
Чтобы использовать это руководство, у вас должны быть установлены сервер Zabbix и веб-интерфейс installed. Вы должны войти в Zabbix под учетной записью пользователя *Admin* по умолчанию.

Применение шаблона мониторинга Linux

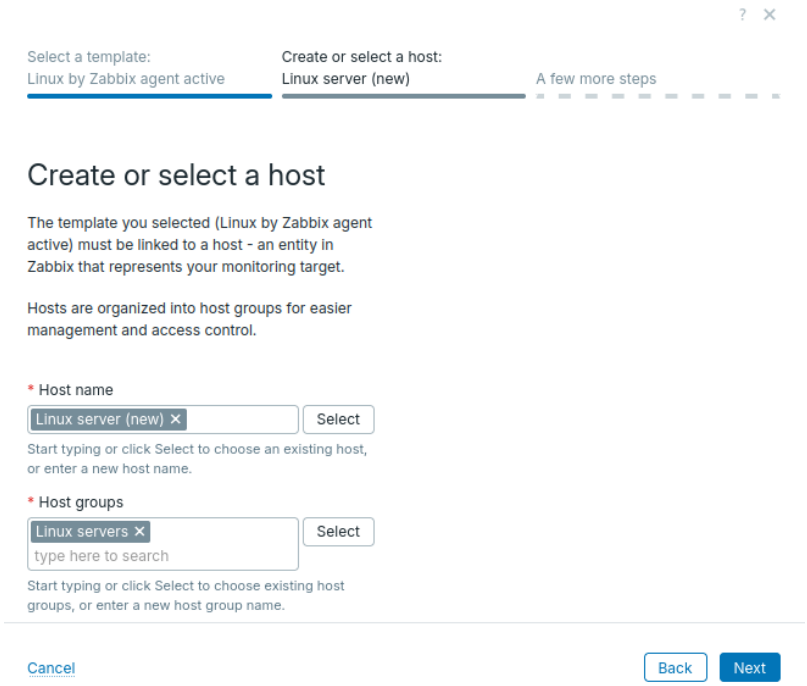
В Zabbix этот шаблон готов к использованию. Он содержит предварительно настроенные элементы, готовые для сбора данных, визуализации и анализа.

Чтобы применить шаблон:

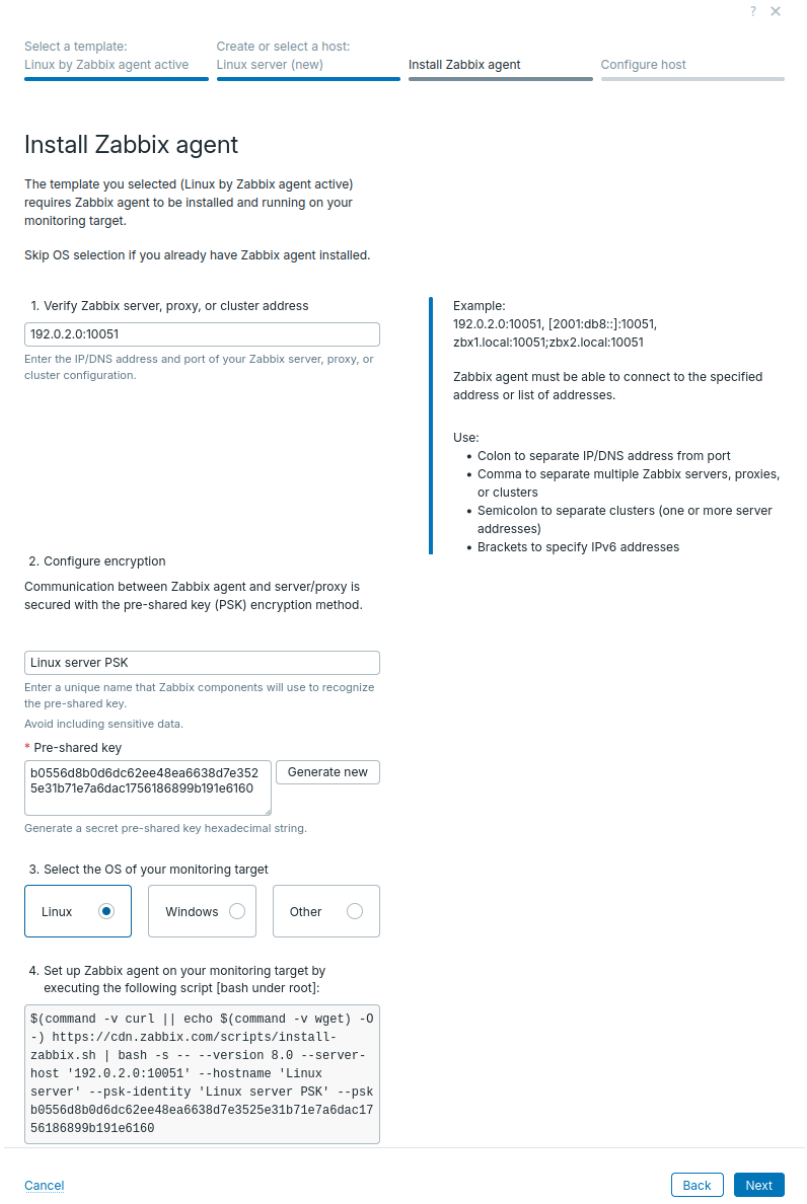
1. Запустите Мастер узла сети в разделе *Сбор данных* > *Узлы сети*.
2. Выберите шаблон *Linux by Zabbix agent active*.



3. Создайте узел сети для Linux-сервера, добавьте для него группу. Этот виртуальный узел сети будет содержать все данные, связанные с мониторингом Linux-сервера.



4. Настройте и установите агент.



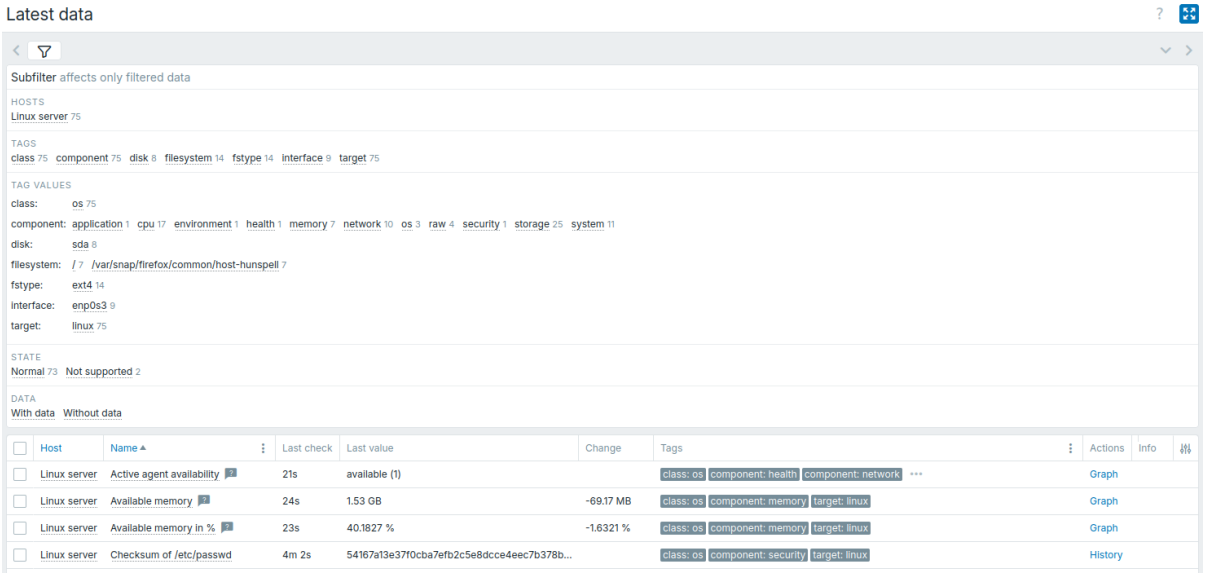
Чтобы установить агент, выполните предоставленную команду на Linux-сервере. Она установит агент и настроит

подключение агента к серверу Zabbix с шифрованием.
 Если агент уже установлен, этот шаг можно пропустить (нажмите *Next*). Однако вам будет предложено указать идентификатор предварительно общего ключа и сам предварительно общий ключ для подключения к агенту.

Остальные параметры можно оставить по умолчанию.

5. Завершите настройку узла сети: нажмите *Create* и *Finish*.

Посмотрите на последние данные



Поздравляем, мониторинг сервера Linux запущен!

Zabbix предоставляет возможности визуализации и отправляет оповещения при возникновении любых проблем. Чтобы настроить оповещения по электронной почте, см. Получение уведомлений о проблемах.

Мониторинг сервера MySQL

Чтобы начать мониторинг сервера MySQL, выполните следующие действия:

1. Создайте ограниченного пользователя MySQL.
2. Примените шаблон мониторинга MySQL.
3. Просмотрите последние данные.

Чтобы начать мониторинг сервера MySQL, вам нужно знать его DSN и IP-адрес сервера Zabbix.

Для мониторинга сервера MySQL будет использоваться агент Zabbix. Инструкции по установке агента будут приведены в процессе.

Чтобы использовать это руководство, у вас должны быть установлены сервер Zabbix и веб-интерфейс installed. Вы должны войти в Zabbix как пользователь *Admin* по умолчанию.

Создание пользователя MySQL с ограниченными правами

Рекомендуется создать пользователя MySQL, права которого ограничены только необходимыми для мониторинга MySQL. Замените <password> на надежный уникальный пароль.

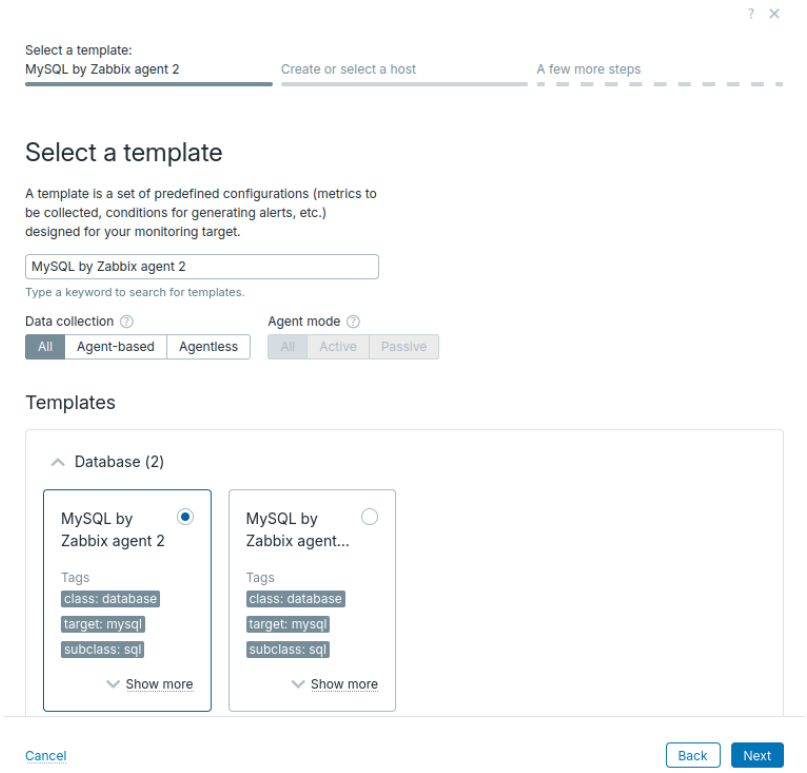
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

Применение шаблона мониторинга MySQL

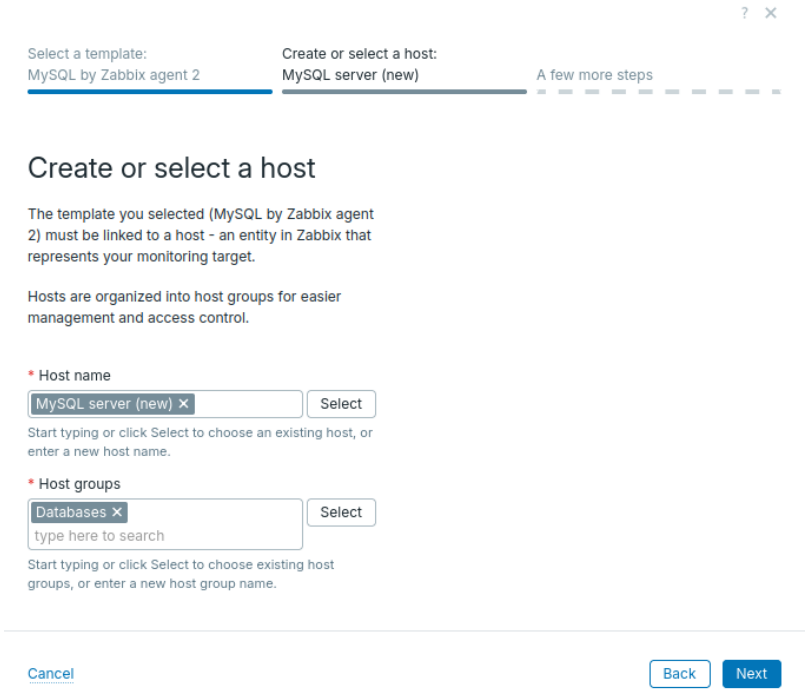
В Zabbix этот шаблон готов к использованию. Он содержит предварительно настроенные элементы, готовые для сбора данных, визуализации и анализа.

Чтобы применить шаблон:

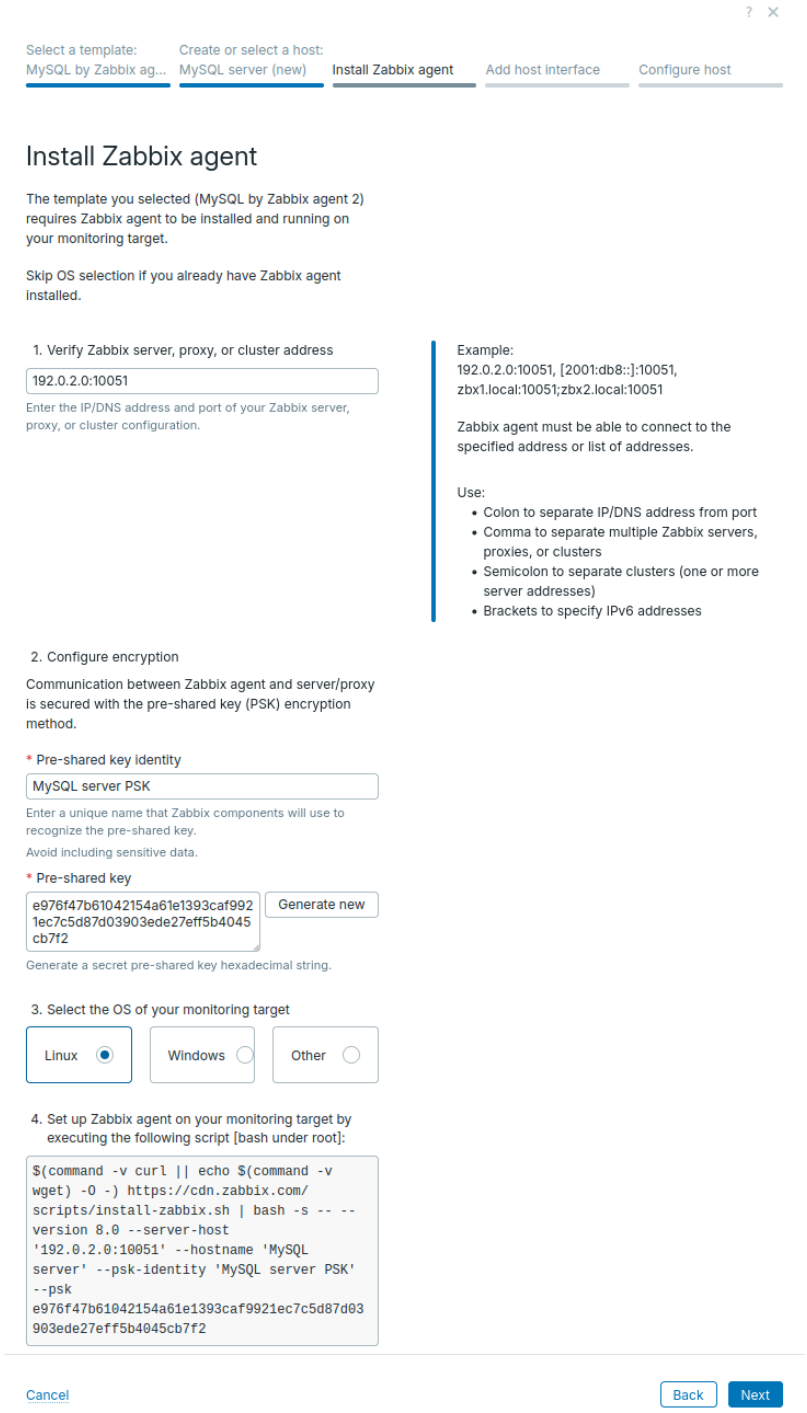
1. Запустите Мастер узла сети в разделе *Сбор данных* > *Узлы сети*.
2. Выберите шаблон *MySQL by Zabbix agent 2*.



3. Создайте узел сети для MySQL server и добавьте для него группу. Этот виртуальный узел сети будет содержать все данные, связанные с мониторингом MySQL server.

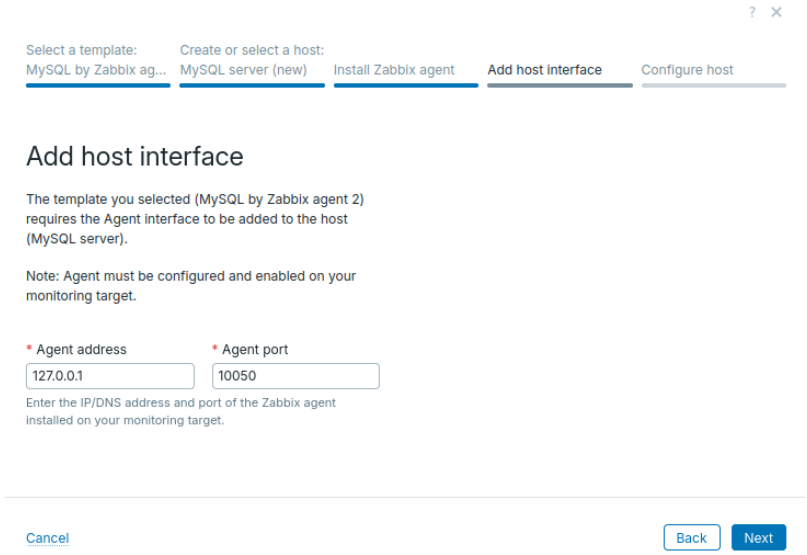


4. Настройте и установите agent 2 (Zabbix agent 2 обеспечивает мониторинг MySQL из коробки).

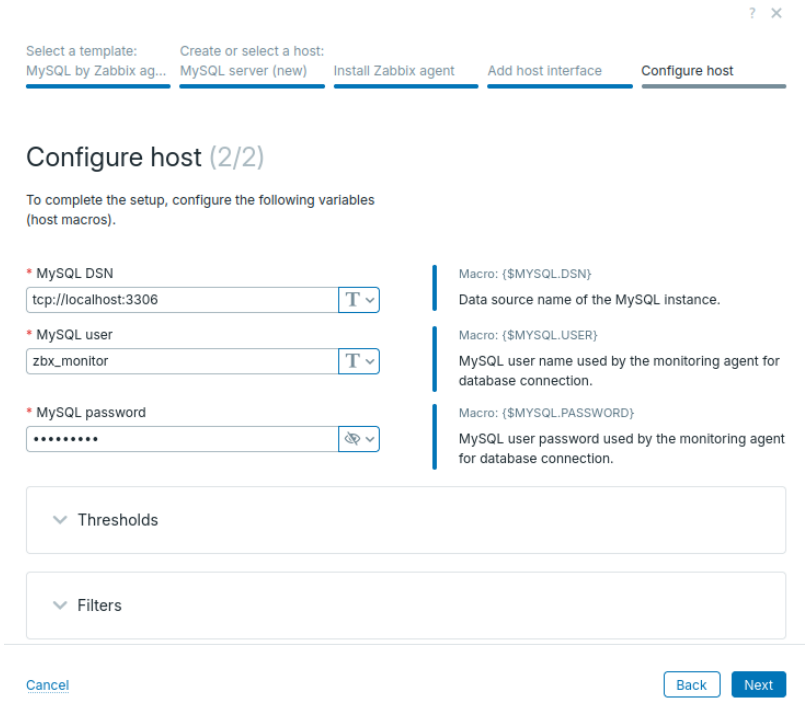


Чтобы установить agent, выполните предоставленную команду на машине, где размещен MySQL server. Она установит agent и настроит подключение agent к серверу Zabbix с шифрованием.
 Если agent 2 уже установлен, этот шаг можно пропустить (нажмите *Далее*). Однако вам будет предложено указать идентификатор предварительно разделяемого ключа и предварительно разделяемый ключ для подключения к agent.

5. Укажите IP-адрес и порт agent (Linux server).



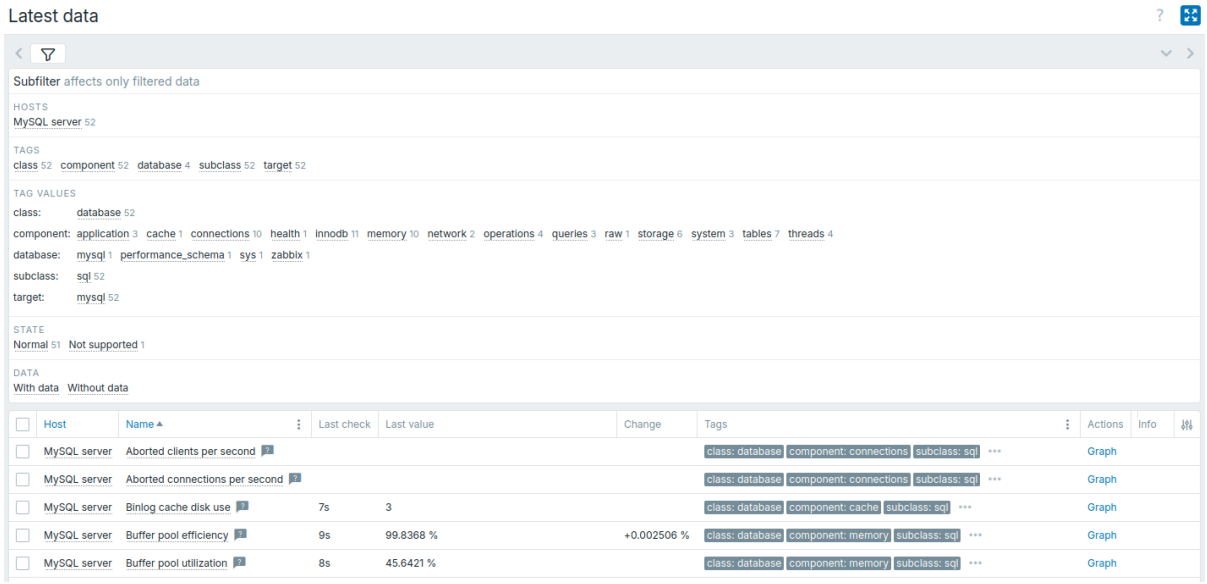
6. Завершите настройку узла сети. Важно: вам будет предложено указать DSN MySQL server, пользователя MySQL и пароль. Укажите пользователя `zbx_monitor` с созданным вами паролем.



Остальные параметры можно оставить по умолчанию.

6. Чтобы завершить работу Мастера узла сети, нажмите *Создать* и *Завершить*.

Просмотр последних данных



Поздравляем, мониторинг сервера MySQL запущен!

Zabbix предоставляет возможности визуализации и отправляет оповещения, если возникают какие-либо проблемы. Чтобы настроить оповещения по электронной почте, см. Получение уведомлений о проблемах.

Дополнительные руководства

Этот раздел документации содержит краткие инструкции по настройке Zabbix для некоторых часто требуемых целей мониторинга.

Он предназначен для новых пользователей Zabbix и может использоваться как навигатор по другим разделам документации, которые содержат информацию, необходимую для решения задачи.

Доступны следующие краткие справочные руководства:

- Мониторинг Linux с помощью агента Zabbix
- Мониторинг Windows с помощью агента Zabbix
- Мониторинг Apache через HTTP
- Мониторинг MySQL с помощью агента Zabbix 2
- Мониторинг VMware с помощью Zabbix
- Мониторинг сетевого трафика с помощью Zabbix
- Мониторинг сетевого трафика с помощью Zabbix с использованием активных проверок
- Мониторинг веб-сайтов с помощью элементов данных Browser
- Мониторинг сертификатов веб-сайтов с помощью агента Zabbix 2 (пассивный режим)
- Мониторинг сетевого коммутатора или маршрутизатора с помощью Zabbix
- Мониторинг журнала событий Windows с использованием активных проверок

Мониторинг Linux с помощью агента Zabbix

Введение

На этой странице вы пройдете через шаги, необходимые для начала базового мониторинга машин Linux с помощью Zabbix. Описанные в этом руководстве шаги можно применить к любой операционной системе на базе Linux.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга вашего Linux-компьютера. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС.

Установка агента Zabbix

Агент Zabbix — это процесс, отвечающий за сбор данных.

Проверьте версию вашего сервера Zabbix:

```
zabbix_server -V
```

Установите агент Zabbix той же версии (рекомендуется) на Linux-машину, которую вы хотите мониторить. В зависимости от ваших потребностей в мониторинге это может быть та же машина, на которой установлен сервер Zabbix, или совершенно другая машина.

Выберите наиболее подходящий способ установки:

- Запустите в контейнере Docker — см. список доступных образов в репозитории Zabbix Docker.
- Установите из пакетов Zabbix (доступно для Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Скомпилируйте из исходных кодов.

Настройка Zabbix для мониторинга

Агент Zabbix может собирать метрики в активном или пассивном режиме (одновременно).

> **Note:**
> Пассивная проверка — это простой запрос данных. Сервер Zabbix или прокси запрашивает некоторые данные (например, загрузку CPU), а агент Zabbix отправляет результат обратно на сервер. Активные проверки требуют более сложной обработки. Сначала агент должен получить от сервера(ов) список элементов данных для независимой обработки, а затем пакетно отправить данные обратно. Дополнительные сведения см. в разделе Пассивные и активные проверки агента.

Шаблоны мониторинга, предоставляемые Zabbix, обычно предлагают два варианта — шаблон для агента Zabbix и шаблон для агента Zabbix (active). В первом случае агент будет собирать метрики в пассивном режиме. Такие шаблоны обеспечивают одинаковые результаты мониторинга, но используют разные протоколы связи.

Дальнейшая настройка Zabbix зависит от того, выберете ли вы шаблон для активных или пассивных проверок агента Zabbix.

Пассивные проверки

Агент Zabbix

1. Откройте файл конфигурации агента на компьютере, где установлен агент.

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

2. Добавьте IP-адрес или DNS-имя (и необязательный порт) вашего сервера Zabbix в параметр сервера. Например:

```
Server=192.0.2.1:10051
```

Агент Zabbix будет использовать этот адрес, чтобы принимать входящие подключения только от указанных серверов Zabbix или прокси для получения данных.

3. Перезапустите агент Zabbix.

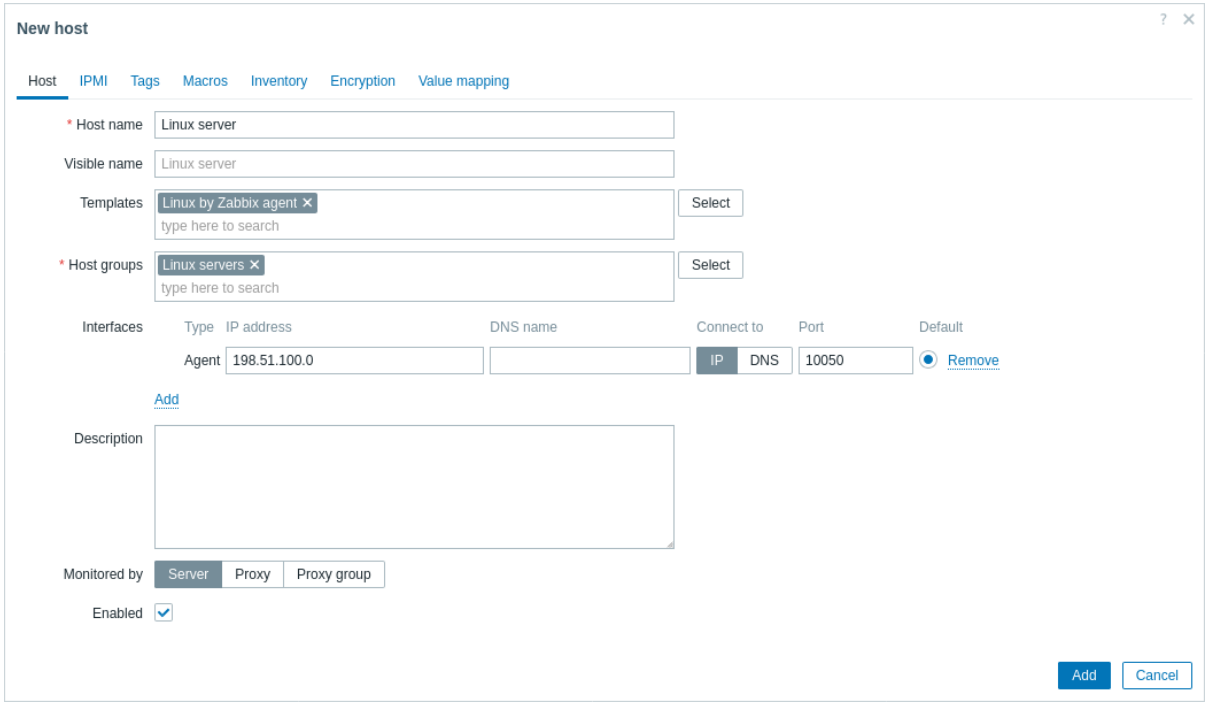
```
systemctl restart zabbix-agent
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

- В поле *Host name* введите имя узла сети (например, ”Linux server”).
- В поле *Templates* введите или выберите шаблон ”Linux by Zabbix agent”, который будет связан с узлом сети.
- В поле *Host groups* введите или выберите группу узлов сети (например, ”Linux servers”).
- В параметре *Interfaces* добавьте интерфейс *Agent* и укажите IP-адрес или DNS-имя машины Linux, на которой установлен агент.



3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять контролируемую машину Linux.

Активные проверки

Агент Zabbix

1. Откройте файл конфигурации агента на машине, где установлен агент:

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

2. Добавьте IP-адрес или DNS-имя (и необязательный порт) вашего сервера Zabbix в параметр ServerActive. Например:

```
ServerActive=192.0.2.1:10051
```

Агент Zabbix будет использовать этот адрес для подключения к порту трапперов сервера Zabbix (по умолчанию: 10051) и запроса данных конфигурации для активных проверок.

3. Определите параметр Hostname, который должен совпадать с именем узла сети, заданным в веб-интерфейсе Zabbix. В нашем примере это:

```
Hostname=Linux server
```

Значение Hostname должно совпадать, поскольку при выполнении активных проверок агент Zabbix использует его для получения правильной конфигурации узла сети с сервера. В частности, агент инициирует подключение к серверу и идентифицирует себя с помощью значения Hostname. Затем сервер предоставляет конфигурацию мониторинга для этого узла сети. Если эти значения различаются, агент не получит соответствующую конфигурацию, что приведет к отсутствию метрик или проблемам с мониторингом.

4. Перезапустите агент Zabbix.

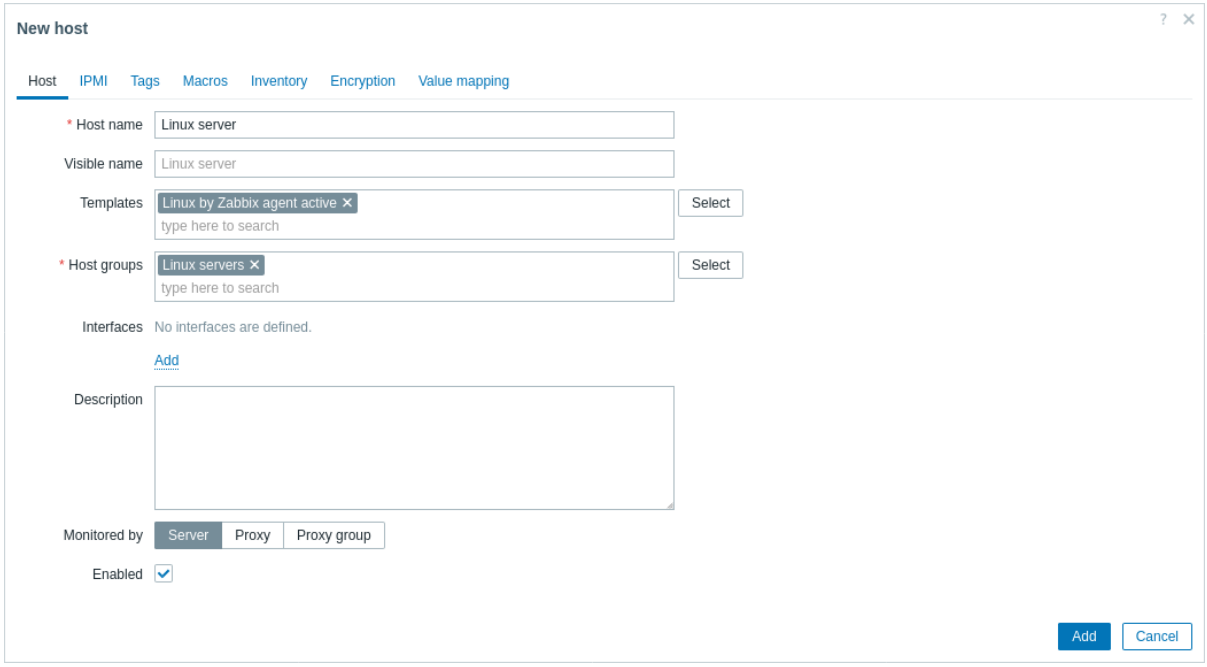
```
systemctl restart zabbix-agent
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

- В поле *Host name* введите имя узла сети (например, ”Linux server”), которое соответствует значению параметра *Hostname*, ранее заданному в файле конфигурации агента.
- В поле *Templates* введите или выберите шаблон ”Linux by Zabbix agent active”, который будет связан с узлом сети.
- В поле *Host groups* введите или выберите группу узлов сети (например, ”Linux servers”).
- Оставьте *Interfaces* не заданным; интерфейс не нужен для активных проверок, поскольку агент инициирует соединение с сервером вместо ожидания соединения от сервера.



3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять контролируемую машину Linux.

Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает ваш Linux-узел сети.

Чтобы просмотреть собранные метрики, откройте раздел меню *Monitoring* > *Hosts* menu section и нажмите *Latest data* рядом с узлом сети.

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

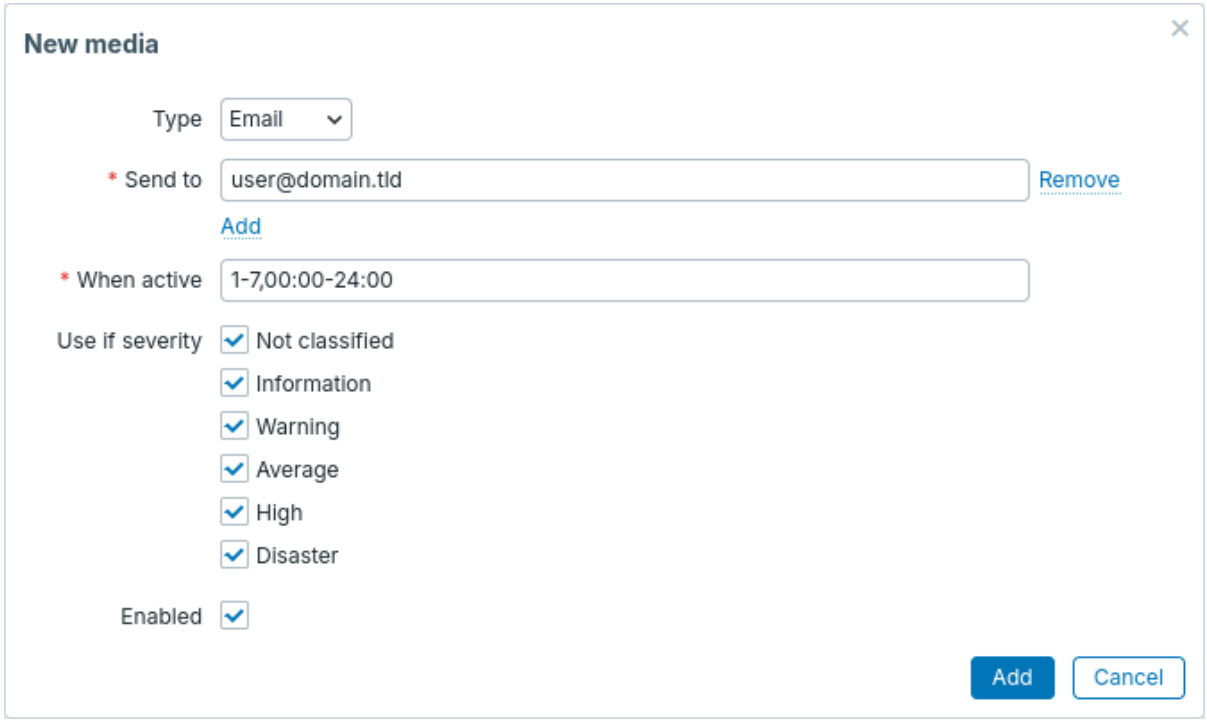
Это действие откроет список всех последних метрик, собранных с узла сети Linux server.

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

Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами.
Это руководство содержит шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *User settings* > *Profile*, откройте вкладку *Media* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомлений о проблемах.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

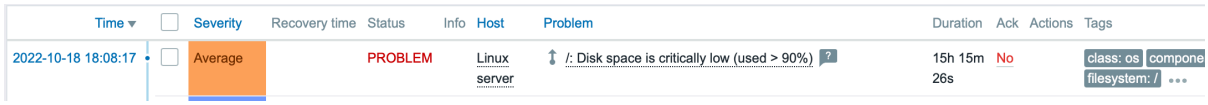
Проверьте вашу конфигурацию

В Linux вы можете имитировать высокую загрузку CPU и, как следствие, получить оповещение о проблеме, выполнив:

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

Возможно, вам потребуется запустить несколько процессов md5sum, чтобы загрузка CPU превысила пороговое значение.

Когда Zabbix обнаружит проблему, она появится в разделе *Monitoring* > *Problems*.



Если оповещения настроены, вы также получите уведомление о проблеме.

См. также:

- Создание элемента данных - как начать мониторинг дополнительных метрик (пользовательский мониторинг без шаблонов).
- Элементы данных Zabbix agent, Элементы данных Zabbix agent для Windows - полный список метрик, которые можно отслеживать с помощью Zabbix agent в Windows.
- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру дата-центра).
- Установка из пакетов - как устанавливать компоненты Zabbix с использованием официальных пакетов RPM и DEB для различных дистрибутивов Linux, обеспечивая доступ к последним функциям и исправлениям ошибок.

Мониторинг Windows с помощью агента Zabbix

Введение

На этой странице описаны шаги, необходимые для начала базового мониторинга узлов сети под управлением Windows с помощью Zabbix.

Для кого предназначено это руководство

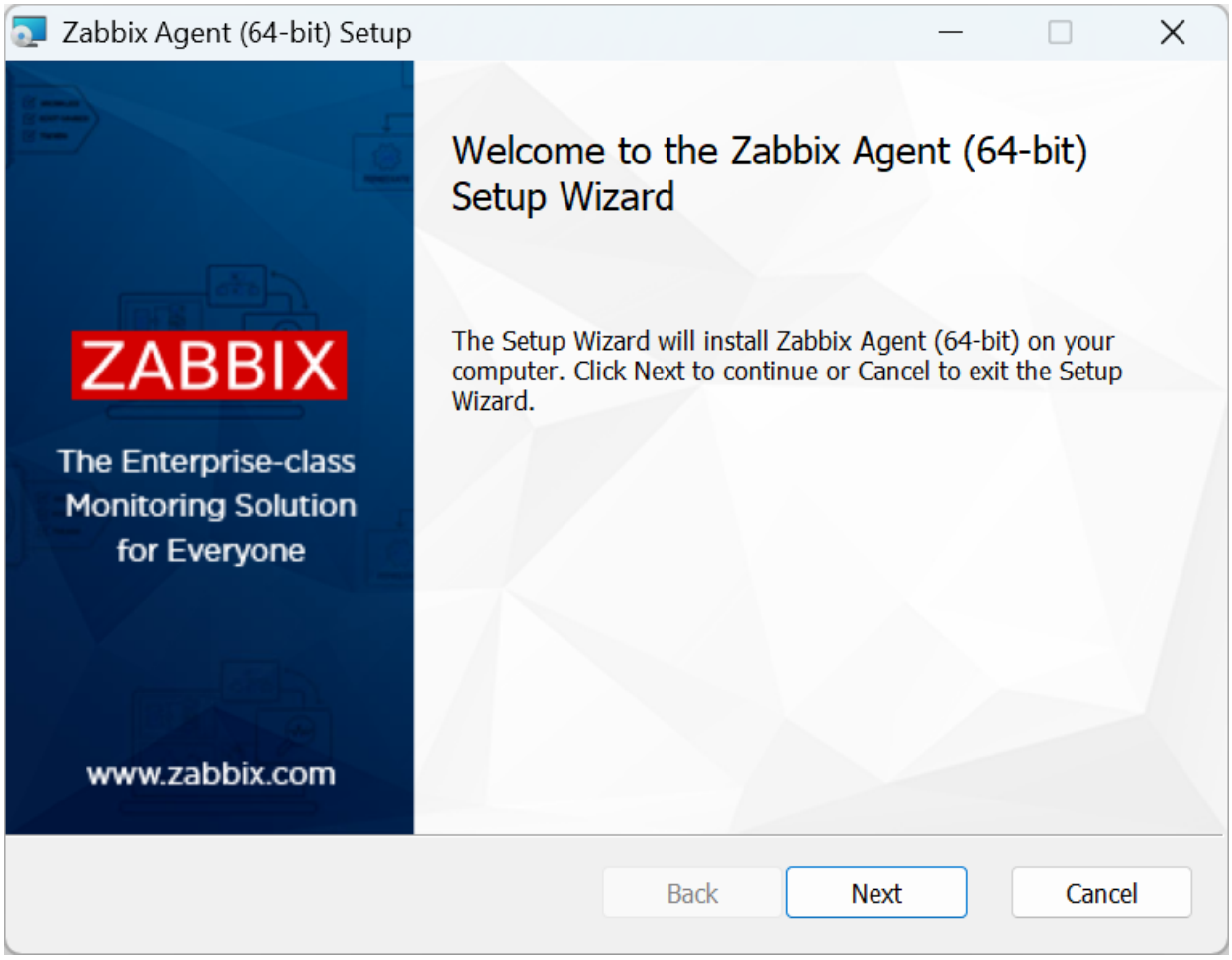
Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга узла сети под управлением Windows. Если вам нужны расширенные возможности настройки или требуется более сложная конфигурация, см. раздел Конфигурация руководства Zabbix.

Предварительные требования

Прежде чем приступить к установке, необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС.

Установка Zabbix агента

Zabbix агент — это процесс, отвечающий за сбор данных. Его необходимо установить на компьютере с Windows, который вы хотите мониторить. Следуйте инструкциям по установке Zabbix агента для Windows.



Настройка Zabbix для мониторинга

Агент Zabbix может собирать метрики в активном или пассивном режиме (одновременно).

Note:
Пассивная проверка — это простой запрос данных. Сервер Zabbix или прокси запрашивает некоторые данные (например, загрузку CPU), а агент Zabbix отправляет результат обратно на сервер. Активные проверки требуют более сложной обработки. Сначала агент должен получить с сервера(ов) список элементов данных для независимой обработки, а затем пакетно отправить данные обратно. Дополнительные сведения см. в разделе Пассивные и активные проверки агента.

Шаблоны мониторинга, предоставляемые Zabbix, обычно предлагают два варианта — шаблон для агента Zabbix и шаблон для агента Zabbix (active). В первом случае агент будет собирать метрики в пассивном режиме. Такие шаблоны обеспечивают одинаковые результаты мониторинга, но используют разные протоколы связи.

Дальнейшая настройка Zabbix зависит от того, выбираете ли вы шаблон для активных или пассивных проверок агента Zabbix.

Пассивные проверки

Веб-интерфейс Zabbix

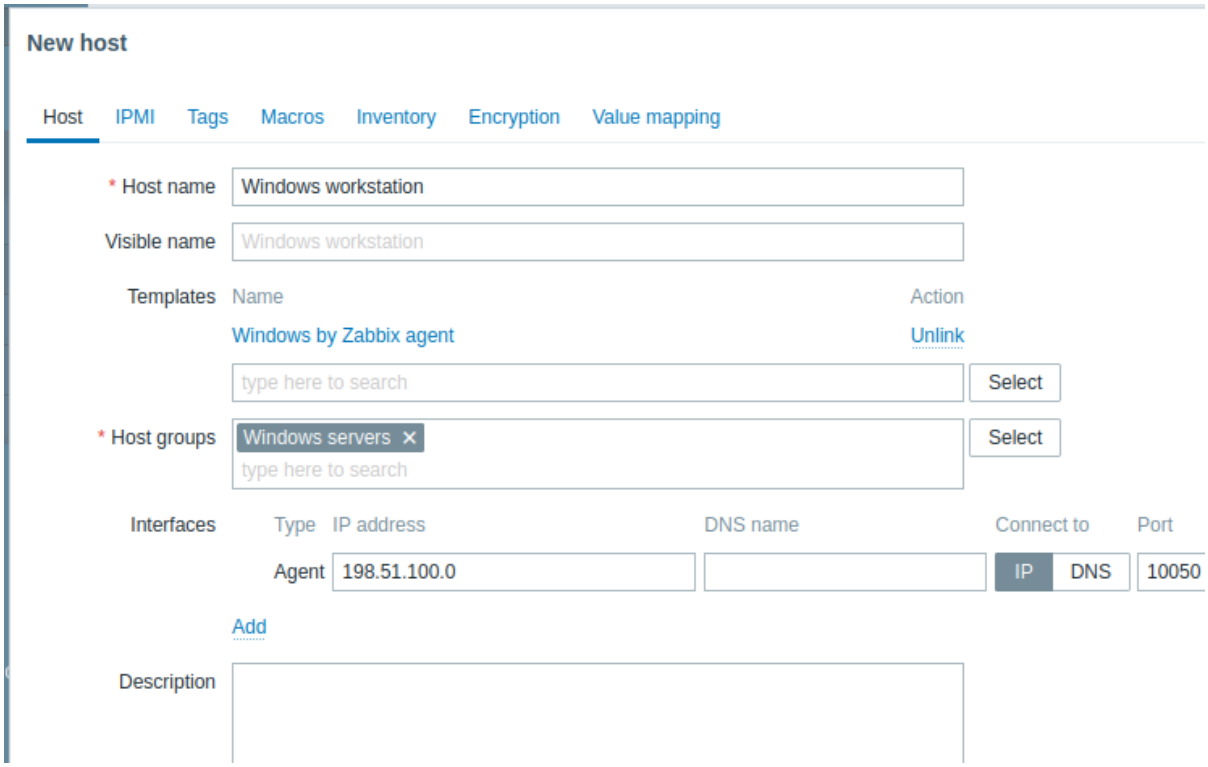
1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

Этот узел сети будет представлять ваш компьютер под управлением Windows.

3. В параметре *Interfaces* добавьте интерфейс *Agent* и укажите IP-адрес или DNS-имя компьютера под управлением Windows, на котором установлен агент.

4. В параметре *Templates* введите или выберите *Windows by Zabbix agent*.



Агент Zabbix

Для пассивных проверок агенту Zabbix необходимо знать IP-адрес или DNS-имя сервера Zabbix. Если во время установки агента вы указали правильные данные, файл конфигурации уже был обновлен. В противном случае необходимо указать их вручную. Откройте файл конфигурации агента и добавьте IP-адрес или DNS-имя вашего сервера Zabbix в параметр `Server`.

Пример:

```
Server=192.0.2.22
```

Активные проверки

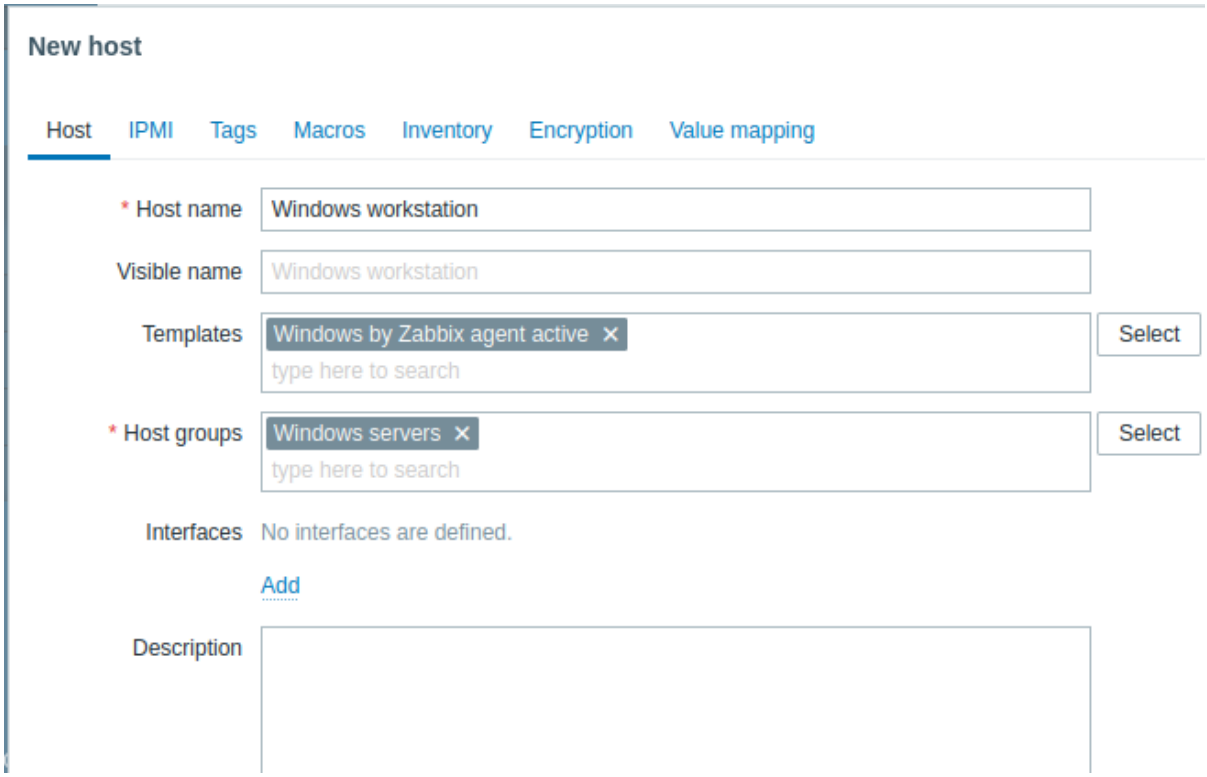
Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

Этот узел сети будет представлять ваш компьютер с Windows.

3. В параметре *Templates* введите или выберите *Windows by Zabbix agent active*.



Агент Zabbix

В папке `C:\Program files\Zabbix Agent` откройте файл `zabbix_agentd.conf` и добавьте:

- Имя узла сети, который вы создали в веб-интерфейсе Zabbix, в параметр *Hostname*.
- IP-адрес или DNS-имя машины с вашим сервером Zabbix в параметр *ServerActive* (может быть заполнен заранее, если вы указали его при настройке агента Zabbix).

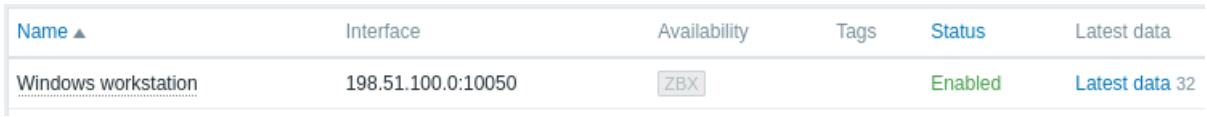
Пример:

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

Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает ваш компьютер с Windows.

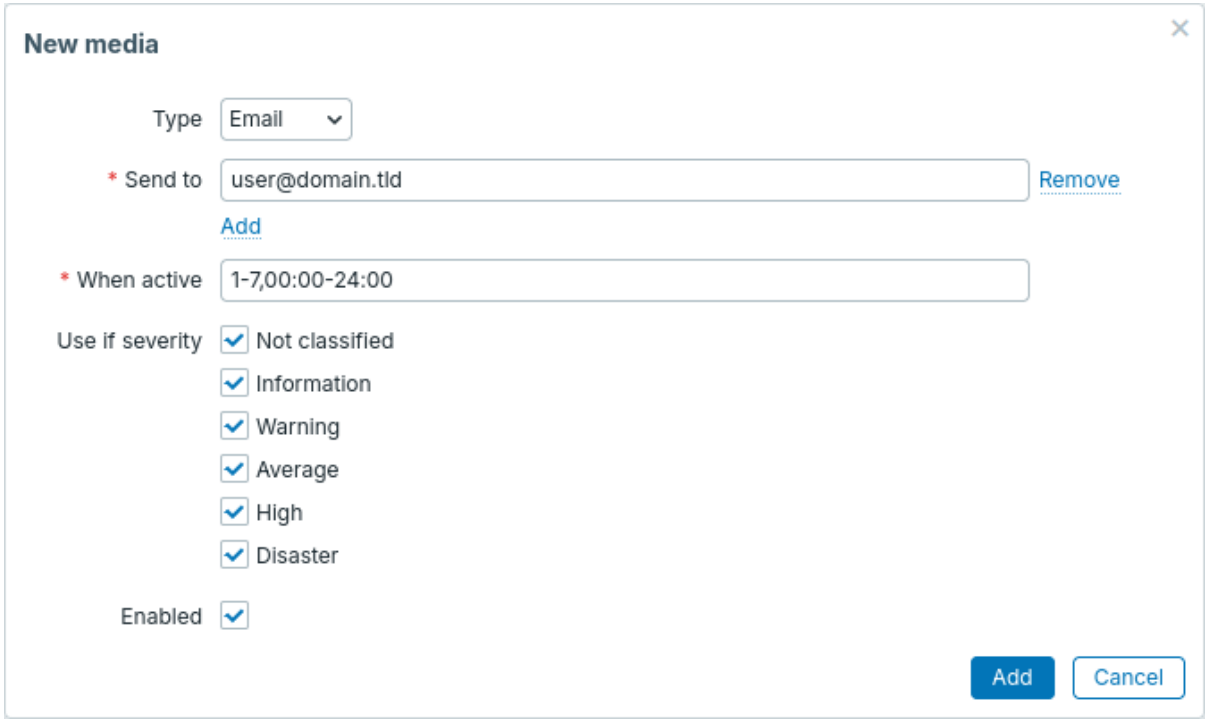
Чтобы просмотреть собранные метрики, откройте раздел меню *Monitoring* > *Hosts* menu section и нажмите *Latest data* рядом с узлом сети.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре с помощью различных методов. В этом руководстве приведены шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *User settings* > *Profile*, откройте вкладку *Media* и добавьте свой email.



2. Следуйте руководству Receiving problem notification.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Note:
В Windows можно использовать утилиту CpuStres, чтобы имитировать высокую загрузку CPU и, как следствие, получить оповещение о проблеме.

См. также:

- Создание элемента данных - как начать мониторинг дополнительных метрик (пользовательский мониторинг без шаблонов).
- Элементы данных агента Zabbix, Элементы данных агента Zabbix для Windows - полный список метрик, которые можно отслеживать с помощью агента Zabbix в Windows.
- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру дата-центра).

Мониторинг Apache через HTTP

Введение

На этой странице показан быстрый и простой способ начать мониторинг веб-сервера Apache без установки какого-либо дополнительного программного обеспечения.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга вашей установки Apache. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить это руководство по установке, необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС.

Подготовка Apache

1. Проверьте, какую версию Apache вы используете:

В системе на базе RHEL выполните:

```
httpd -v
```

В Debian/Ubuntu выполните:

```
apache2 -v
```

2. Убедитесь, что модуль Status включен в вашем экземпляре Apache.

В системе на базе RHEL выполните:

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

В Debian/Ubuntu выполните:

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

Если в списке нет status_module, включите модуль, выполнив:

В системе на базе RHEL выполните:

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

В Debian/Ubuntu выполните:

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

3. Отредактируйте файл конфигурации Apache, чтобы разрешить доступ к отчетам о состоянии с IP-адреса сервера Zabbix.

В системе на базе RHEL: `/etc/httpd/conf.modules.d/status.conf`:

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

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

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

Добавьте в файл следующие строки (**замените 198.51.100.255** на IP-адрес вашего сервера Zabbix):

- Для Apache 2.2:
  <Location /server-status> SetHandler server-status

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

4. Перезапустите Apache

В системе на базе RHEL выполните:

```
sudo systemctl restart httpd
```

В Debian/Ubuntu выполните:

```
sudo systemctl restart apache2
```

5. Чтобы проверить, что все настроено правильно, выполните (**замените 198.51.100.255** на IP-адрес вашего сервера Zabbix):

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

Ответ должен содержать статистику веб-сервера Apache.

Настройка Zabbix для мониторинга

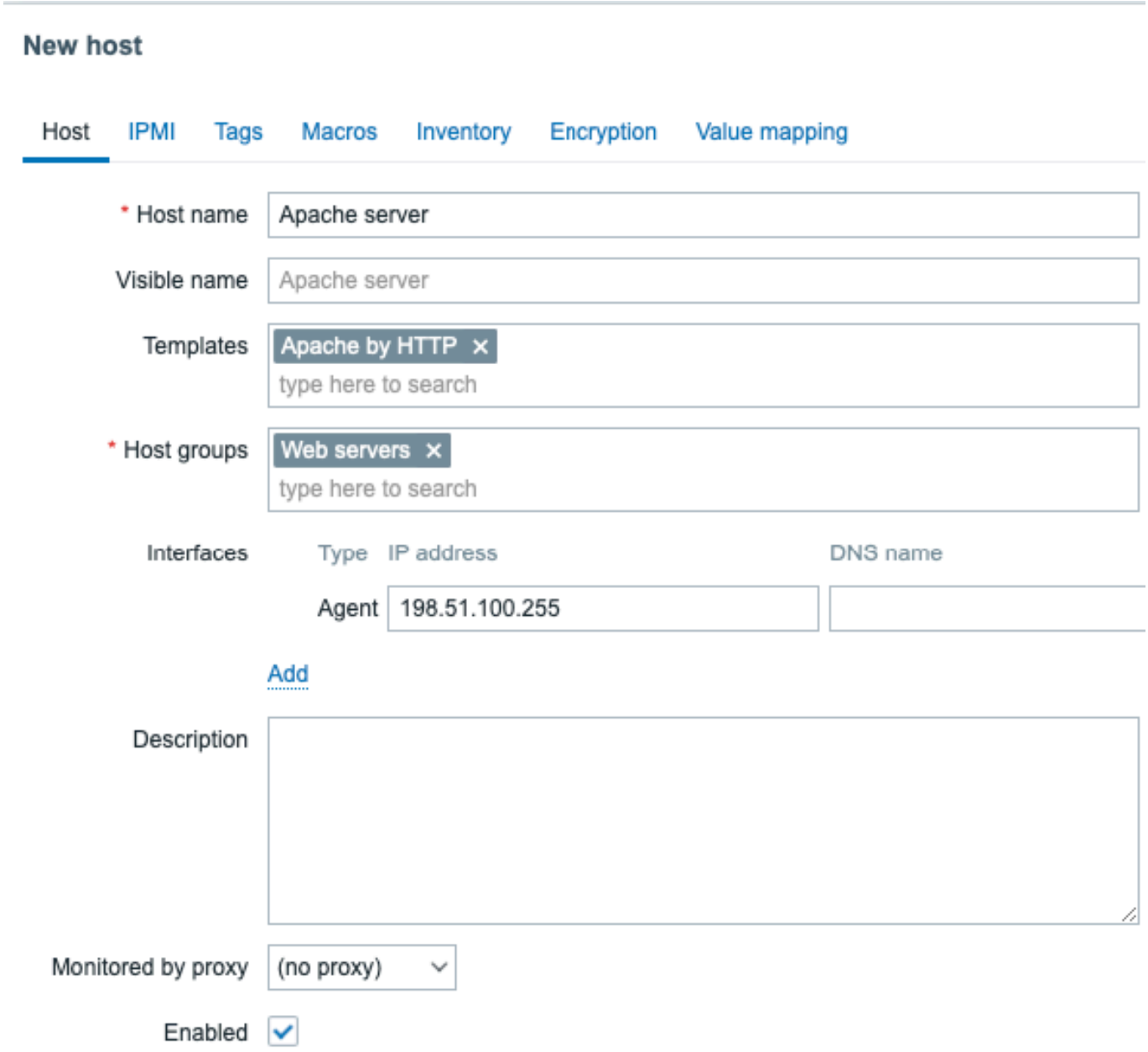
1. Войдите во веб-интерфейс Zabbix.

2. Создайте узел сети во веб-интерфейсе Zabbix.

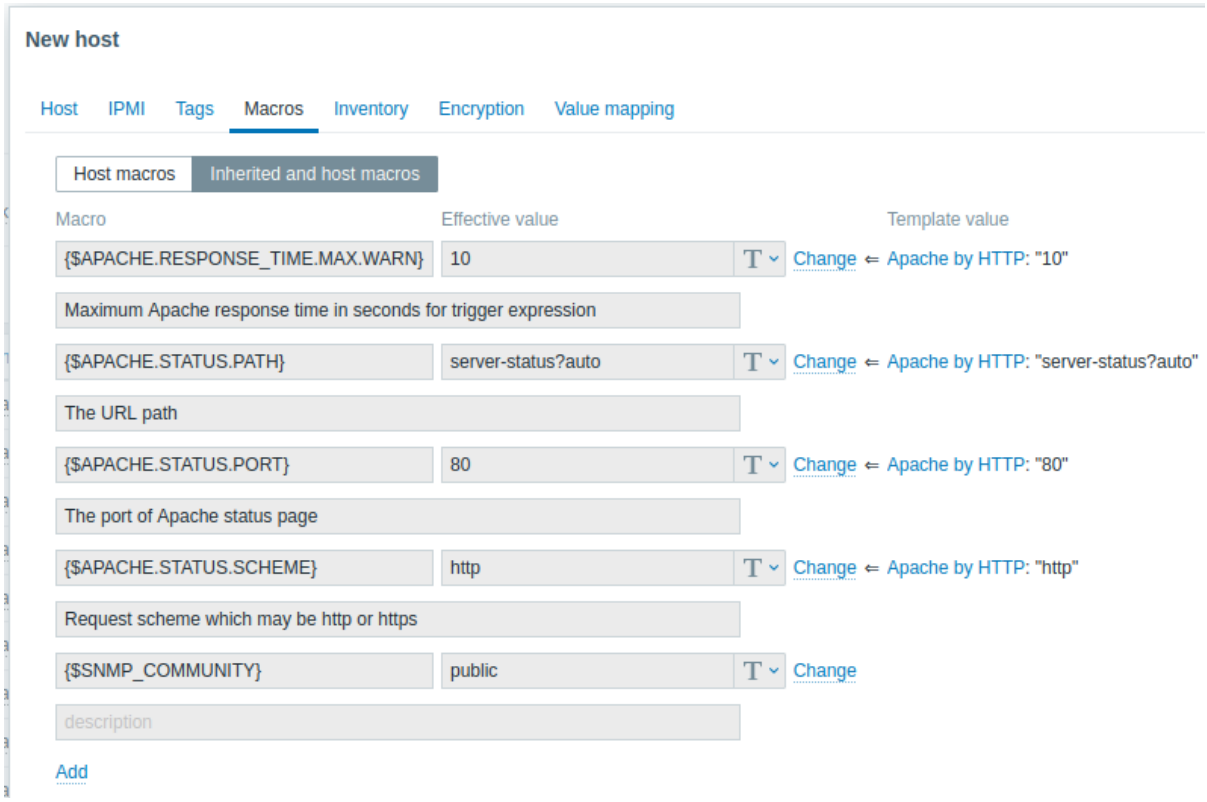
Этот узел сети будет представлять ваш сервер Apache.

3. В параметре *Interfaces* добавьте интерфейс *агент* и укажите IP-адрес вашего экземпляра Apache. **Вам не нужно устанавливать агент Zabbix на машине**, интерфейс будет использоваться только для разрешения макроса {HOST.CONN}. Этот макрос используется в элементах данных шаблона для определения местоположения экземпляра Apache.

4. В параметре *Templates* введите или выберите *Apache by HTTP*.



5. Перейдите на вкладку **Macros** и выберите режим *Inherited and host macros*. Проверьте, что значения макросов {$APACHE.STATUS.PORT} и {$APACHE.STATUS.SCHEME} соответствуют настройкам вашей установки. По умолчанию порт — 80, а схема — http. Измените значения макросов, если вы используете другой порт и/или схему.



Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает ваш веб-сервер Apache.

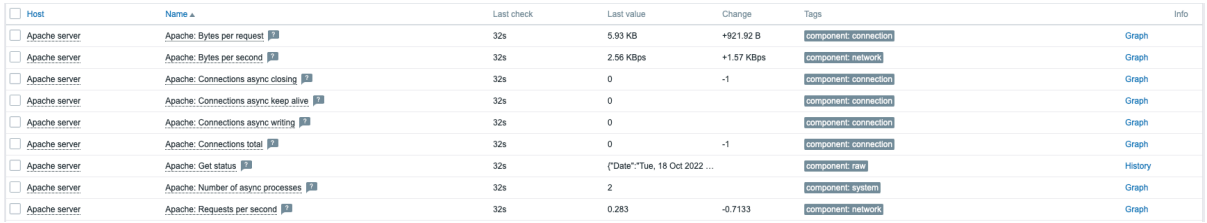
Чтобы просмотреть собранные метрики, откройте раздел меню *Monitoring* > *Hosts* menu section и нажмите *Dashboards* рядом с узлом сети.



Это действие откроет панель узла сети с наиболее важными метриками, собранными со страницы Apache /server-status.



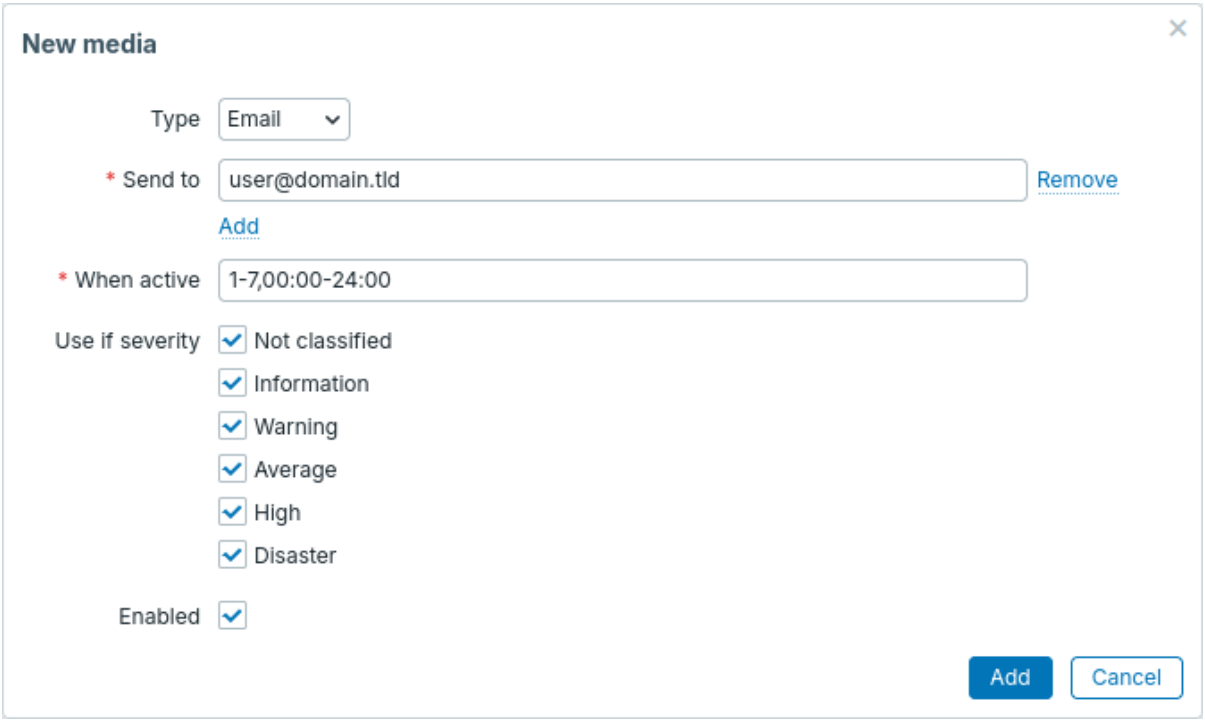
Кроме того, в разделе *Monitoring* > *Hosts* можно нажать *Latest data*, чтобы просмотреть все последние собранные метрики в виде списка.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре с помощью различных методов.
Это руководство содержит шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *User settings* > *Profile*, откройте вкладку *Media* и добавьте свой email.



2. Следуйте руководству по получению уведомлений о проблемах.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию

Чтобы смоделировать реальную проблему и получить тестовое оповещение о проблеме:

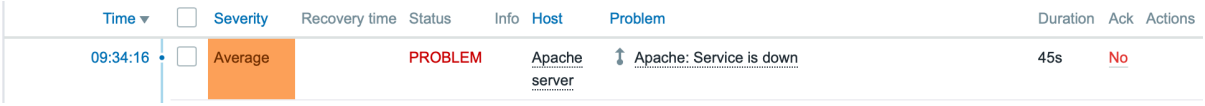
1. Откройте конфигурацию узла сети *Apache server* в Zabbix.

2. Перейдите на вкладку Macros и выберите *Inherited and host macros*.

3. Нажмите *Change* рядом с макросом {$APACHE.STATUS.PORT} и задайте другой порт.

4. Нажмите *Update*, чтобы сохранить конфигурацию узла сети.

5. Через несколько минут Zabbix обнаружит проблему *Apache service is down*, потому что теперь он не может подключиться к экземпляру. Она появится в разделе *Monitoring* > *Problems*.



Если оповещения настроены, вы также получите уведомление о проблеме.

6. Измените значение макроса обратно, чтобы устранить проблему и продолжить мониторинг Apache.

См. также:

- Усиление защиты веб-сервера - рекомендуемые настройки для повышения безопасности веб-сервера.
- Создание элемента данных - как начать мониторинг дополнительных метрик.
- Элементы данных HTTP - как отслеживать пользовательские метрики с помощью HTTP-агента.
- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, а затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).

Мониторинг MySQL с помощью агента Zabbix 2

Введение

На этой странице описаны шаги, необходимые для начала базового мониторинга сервера MySQL.

Для мониторинга сервера MySQL существует несколько подходов: агент Zabbix, агент Zabbix 2 или стандарт Open Database Connectivity (ODBC). Основное внимание в этом руководстве уделено мониторингу сервера MySQL с помощью агента Zabbix 2, который является **рекомендуемым** подходом благодаря простой настройке в различных конфигурациях. Однако на этой странице также приведены инструкции для других подходов, поэтому вы можете выбрать тот, который лучше всего соответствует вашим требованиям.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга сервера MySQL. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо загрузить и установить сервер Zabbix, веб-интерфейс Zabbix и агент Zabbix 2 в соответствии с инструкциями для вашей ОС.

В зависимости от вашей среды некоторые шаги в этом руководстве могут незначительно отличаться. Это руководство основано на среде, работающей под управлением Ubuntu.

Создание пользователя MySQL

Для мониторинга сервера MySQL Zabbix требуется доступ к нему и его процессам. В вашей установке MySQL уже есть пользователь с необходимым уровнем доступа (пользователь "zabbix", который был создан при установке Zabbix), однако у этого пользователя больше привилегий, чем необходимо для простого мониторинга (привилегии на DROP баз данных, DELETE записей из таблиц и т. д.). Поэтому для *только* мониторинга сервера MySQL необходимо создать отдельного пользователя MySQL.

1. Подключитесь к клиенту MySQL, создайте пользователя "zbx_monitor" (замените *<password>* для пользователя "zbx_monitor" на пароль по вашему выбору) и GRANT необходимые привилегии этому пользователю:

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

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

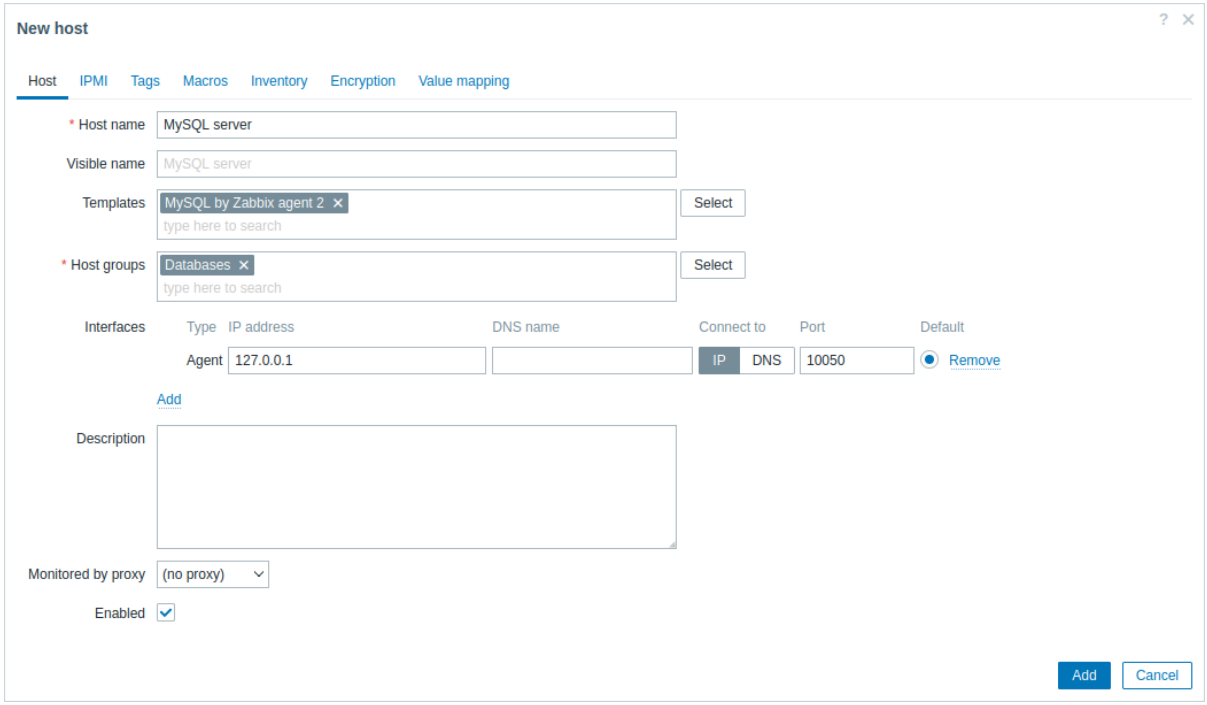
После создания пользователя можно перейти к следующему шагу.

Настройка веб-интерфейса Zabbix

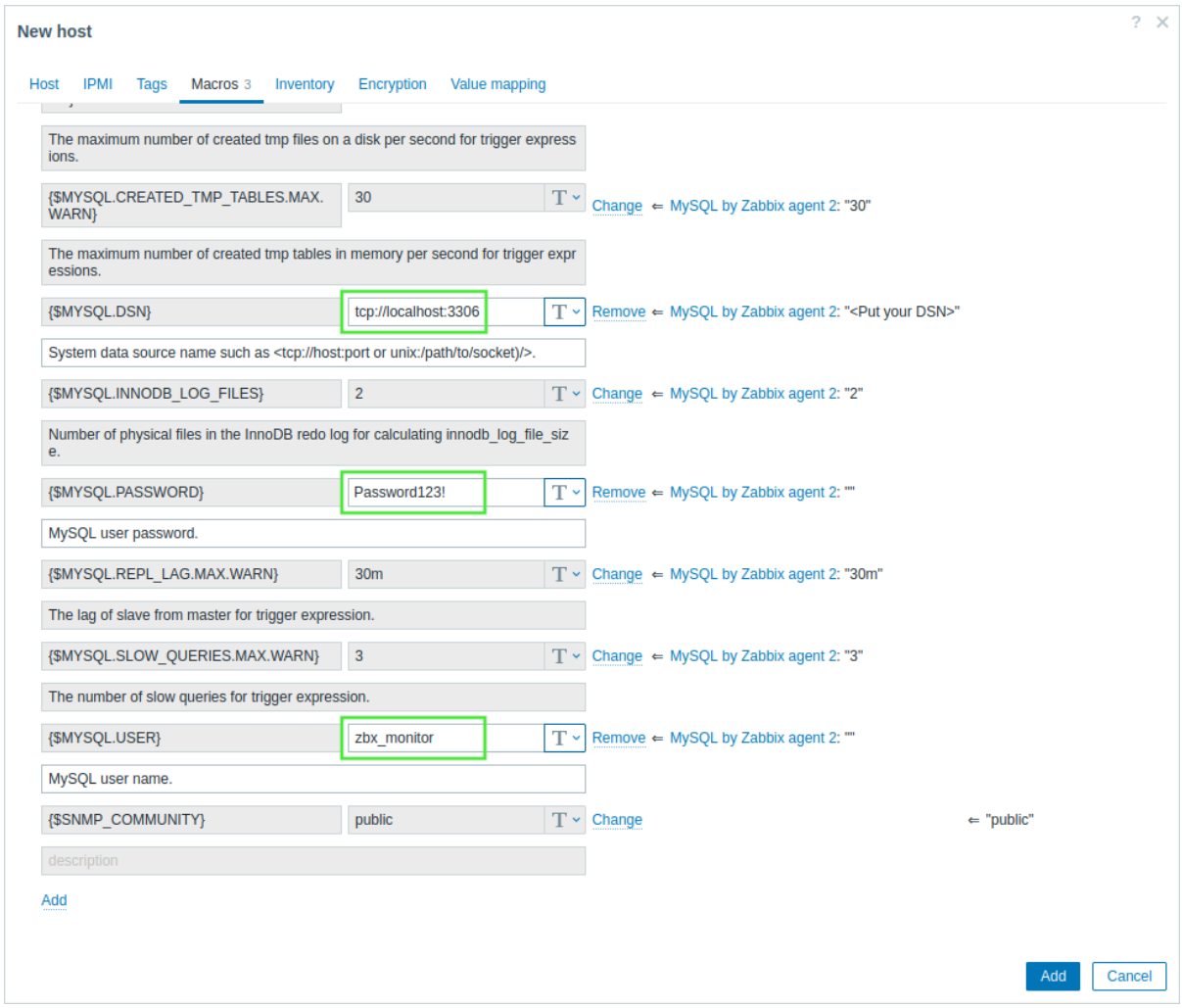
1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix:

- В поле *Host name* введите имя узла сети (например, "MySQL server").
- В поле *Templates* введите или выберите шаблон "MySQL by Zabbix agent 2", который будет связан с узлом сети.
- В поле *Host groups* введите или выберите группу узлов сети (например, "Databases").
- В поле *Interfaces* добавьте интерфейс типа "Agent" и укажите IP-адрес вашего MySQL server. В этом руководстве используется "127.0.0.1" (localhost) для мониторинга MySQL server, установленного на той же машине, что и сервер Zabbix и агент Zabbix agent 2.



- На вкладке *Macros* переключитесь на *Inherited and host macros*, найдите следующие макросы и нажмите *Change* рядом со значением макроса, чтобы обновить его:
  - {$MYSQL.DSN} - задайте источник данных MySQL server (строку подключения именованной сессии из файла конфигурации плагина MySQL для агента Zabbix agent 2). В этом руководстве используется источник данных по умолчанию ”tcp://localhost:3306” для мониторинга MySQL server, установленного на той же машине, что и сервер Zabbix и агент Zabbix agent 2.
  - {$MYSQL.PASSWORD} - задайте пароль ранее созданного пользователя MySQL ”zbx_monitor”.
  - {$MYSQL.USER} - задайте имя ранее созданного пользователя MySQL ”zbx_monitor”.

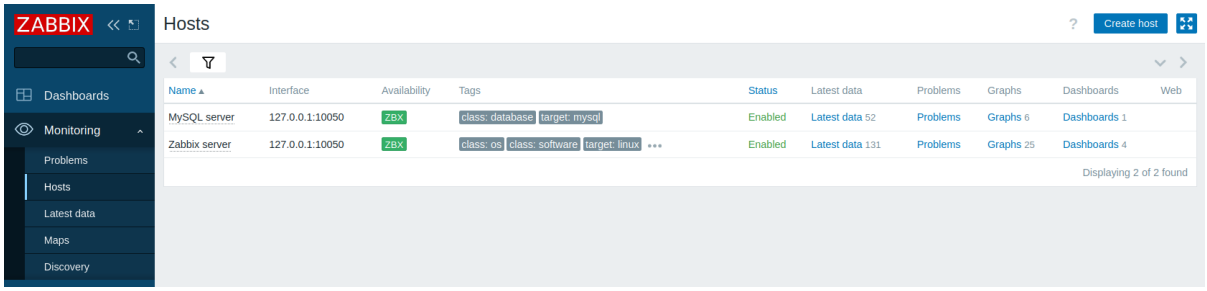


3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять ваш MySQL server.

Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает ваш сервер MySQL.

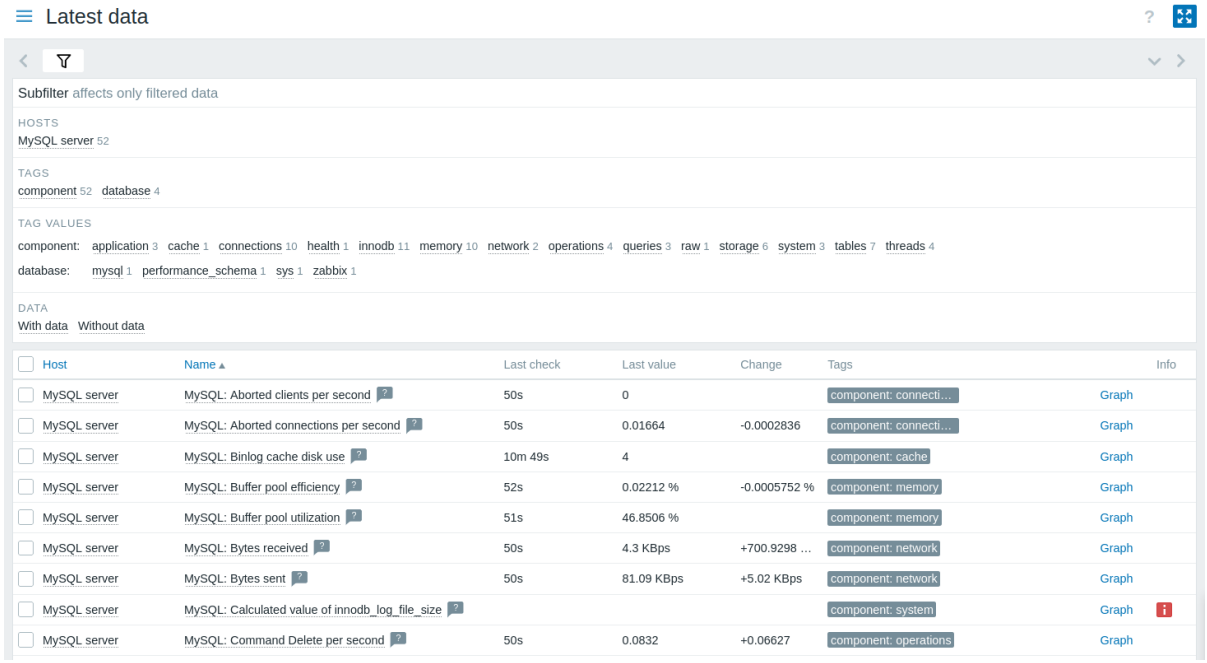
Чтобы просмотреть собранные метрики, перейдите в раздел меню *Мониторинг > Узлы сети* и нажмите *Дашборды* рядом с узлом сети.



Это действие откроет дашборд узла сети (настроенный на уровне шаблона) с наиболее важными метриками, собранными с сервера MySQL.



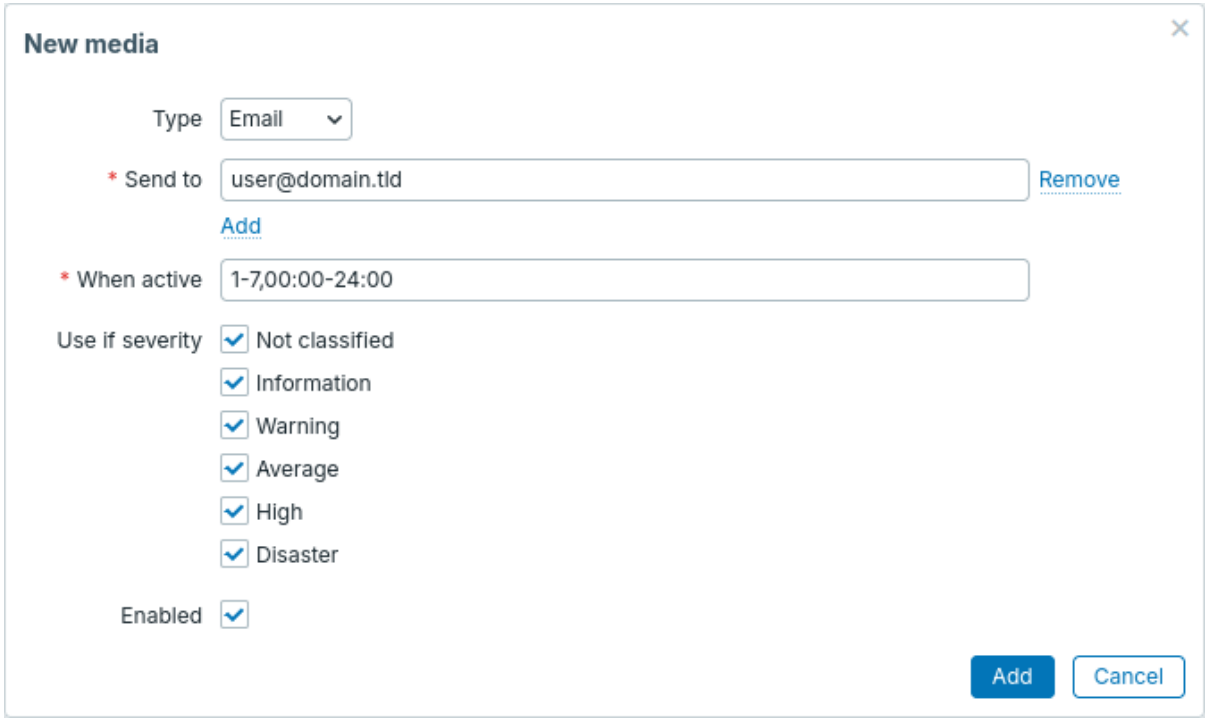
Кроме того, в разделе меню *Мониторинг > Узлы сети* вы можете нажать *Последние данные*, чтобы просмотреть все последние собранные метрики в виде списка. Обратите внимание, что у элемента данных *MySQL: Calculated value of innodb_log_file_size* не ожидается наличие данных, так как значение будет вычисляться на основе данных за последний час.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре с помощью различных методов. Это руководство содержит основные шаги настройки для отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя > Профиль*, откройте вкладку *Медиа* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию

Чтобы проверить вашу конфигурацию, мы можем смоделировать реальную проблему, изменив конфигурацию узла сети в веб-интерфейсе Zabbix.

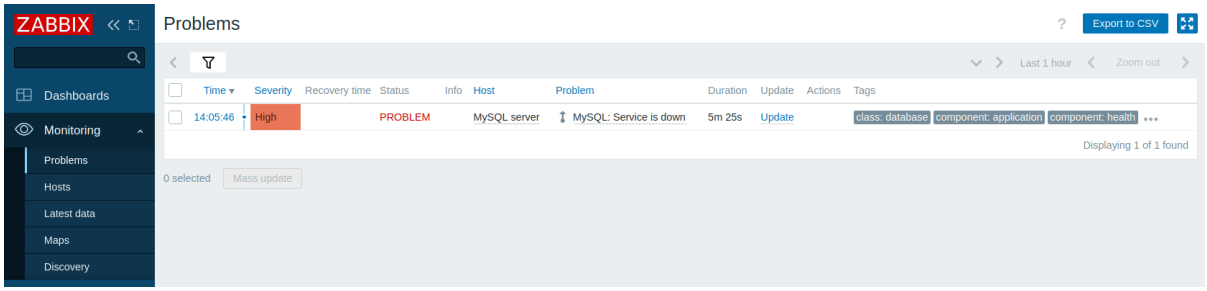
1. Откройте конфигурацию узла сети MySQL server в Zabbix.

2. Перейдите на вкладку *Macros* и выберите *Inherited and host macros*.

3. Нажмите *Change* рядом, например, со значением макроса {$MYSQL.USER}, настроенного ранее, и задайте другое имя пользователя MySQL.

4. Нажмите *Update*, чтобы обновить конфигурацию узла сети.

5. Через несколько мгновений Zabbix обнаружит проблему ”MySQL: Service is down”, поскольку не сможет подключиться к серверу MySQL. Проблема появится в *Monitoring > Problems*.



Если оповещения настроены, вы также получите уведомление о проблеме.

6. Измените значение макроса обратно на прежнее, чтобы устранить проблему и продолжить мониторинг сервера MySQL.

Другие способы мониторинга MySQL

Вместо мониторинга сервера MySQL с помощью агента Zabbix 2 можно также использовать агент Zabbix или стандарт Open Database Connectivity (ODBC). Хотя рекомендуется использовать агент Zabbix 2, могут существовать некоторые конфигурации, которые не поддерживают агент Zabbix 2 или требуют нестандартного подхода.

Ключевое различие между агентом Zabbix и ODBC заключается в методе сбора данных: агент Zabbix устанавливается непосредственно на сервер MySQL и собирает данные с помощью встроенных возможностей, тогда как ODBC использует драйвер ODBC для установления соединения с сервером MySQL и получения данных с помощью SQL-запросов.

Хотя многие шаги настройки похожи на мониторинг сервера MySQL с помощью агента Zabbix 2, есть и некоторые существенные различия — для мониторинга сервера MySQL необходимо настроить агент Zabbix или ODBC. Следующие инструкции помогут вам разобраться в этих **различиях**.

Мониторинг MySQL с помощью агента Zabbix

Чтобы отслеживать сервер MySQL с помощью агента Zabbix, необходимо скачать и установить сервер Zabbix, веб-интерфейс Zabbix и агент Zabbix в соответствии с инструкциями для вашей ОС.

После успешной установки необходимых компонентов Zabbix нужно создать пользователя MySQL, как описано в разделе *Создание пользователя MySQL*.

После создания пользователя MySQL необходимо настроить агент Zabbix так, чтобы он мог устанавливать соединение с сервером MySQL и отслеживать его. Это включает настройку нескольких пользовательских параметров для выполнения пользовательских проверок агента, а также предоставление агенту Zabbix необходимых учетных данных для подключения к серверу MySQL от имени ранее созданного пользователя ”zbx_monitor”.

Настройка агента Zabbix

1. Перейдите в каталог дополнительных конфигураций агента Zabbix.

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

> **Attention:**
> Каталог дополнительных конфигураций агента Zabbix должен находиться в том же каталоге, что и файл конфигурации агента Zabbix; путь по умолчанию зависит от ОС и способа установки Zabbix.

Вместо определения всех необходимых пользовательских параметров для мониторинга сервера MySQL в файле конфигурации агента Zabbix эти параметры будут определены в отдельном файле в каталоге дополнительных конфигураций.

2. Создайте файл `template_db_mysql.conf` в каталоге дополнительных конфигураций агента Zabbix.

```
vi template_db_mysql.conf
```

3. Скопируйте содержимое файла `template_db_mysql.conf`, расположенного в репозитории Zabbix, в созданный файл `template_db_mysql.conf` и сохраните его.

4. Перезапустите агент Zabbix, чтобы обновить его конфигурацию.

```
systemctl restart zabbix-agent
```

После настройки пользовательских параметров агента Zabbix можно перейти к настройке учетных данных, которые позволят агенту Zabbix получать доступ к серверу MySQL.

5. Перейдите в домашний каталог агента Zabbix (если он не существует в вашей системе, его нужно создать; по умолчанию: */var/lib/zabbix*).

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

6. Создайте файл *.my.cnf* в домашнем каталоге агента Zabbix.

```
vi .my.cnf
```

7. Скопируйте следующий содержимое в файл *.my.cnf* (замените <*password*> на пароль пользователя ”zbx_monitor”).

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

Настройте веб-интерфейс Zabbix и проверьте вашу конфигурацию

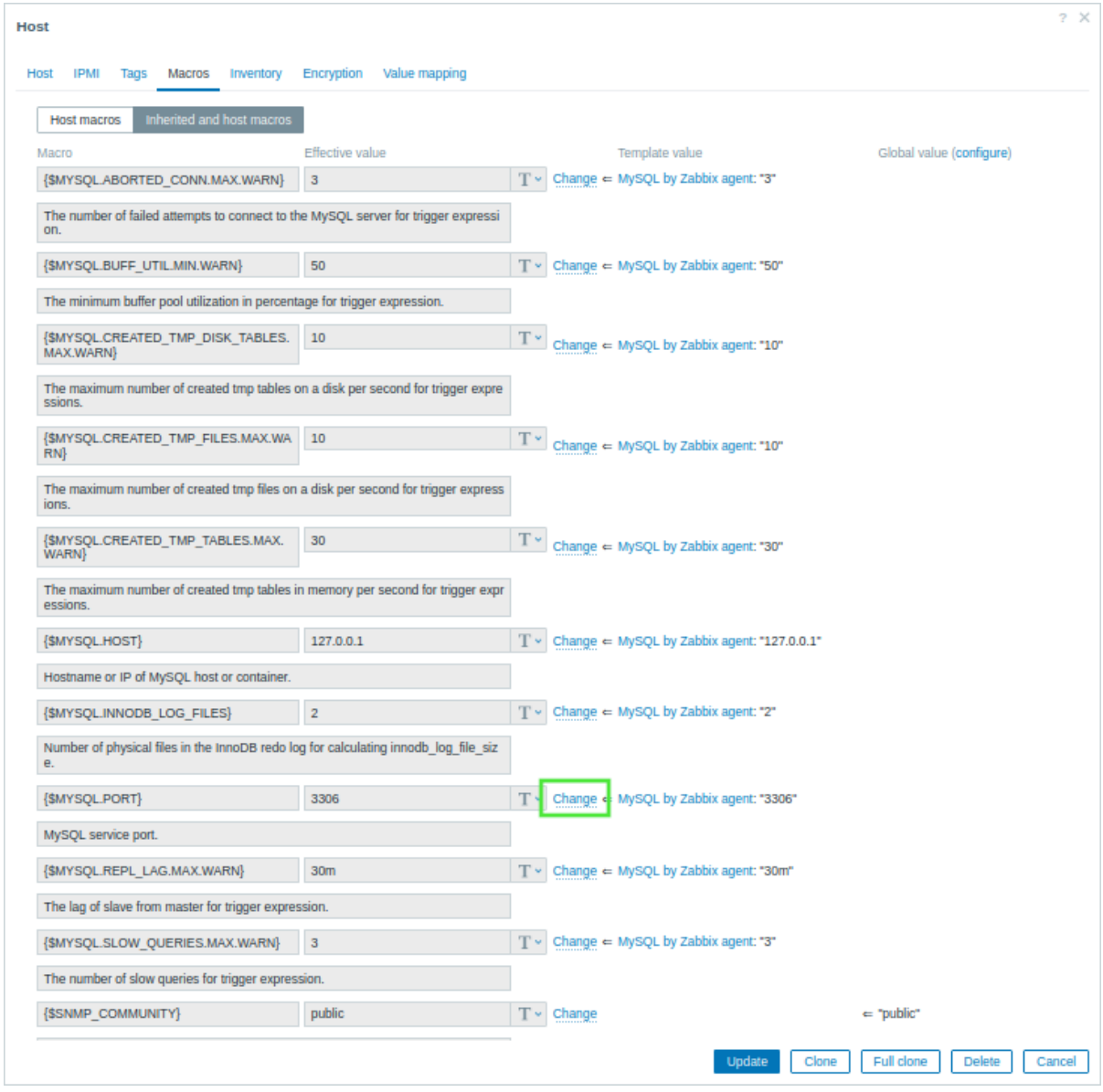
Чтобы настроить веб-интерфейс Zabbix, следуйте инструкциям в разделе *Настройка веб-интерфейса Zabbix* с учетом следующих изменений:

- В поле *Шаблоны* введите или выберите шаблон ”MySQL by Zabbix agent”, который будет привязан к узлу сети.
- Настройка *Макросов* не требуется.

После настройки веб-интерфейса Zabbix вы сможете просматривать собранные метрики и настраивать оповещения о проблемах.

Чтобы проверить вашу конфигурацию, следуйте инструкциям в разделе *Проверка вашей конфигурации* с учетом следующих изменений:

- В разделе *Унаследованные макросы и макросы узла сети* конфигурации узла сети MySQL server нажмите *Изменить* рядом со значением макроса {$MYSQL.PORT} и задайте другой порт (например, ”6033”).



Мониторинг MySQL с помощью ODBC

Чтобы мониторить сервер MySQL с помощью ODBC, необходимо скачать и установить сервер Zabbix и веб-интерфейс Zabbix.

После успешной установки необходимых компонентов Zabbix нужно создать пользователя MySQL, как описано в разделе *Создание пользователя MySQL*.

После создания пользователя MySQL необходимо настроить ODBC. Это включает установку одной из наиболее часто используемых реализаций API ODBC с открытым исходным кодом — unixODBC — и драйвера unixODBC, а также редактирование файла конфигурации драйвера ODBC.

Настройка ODBC

1. Установите unixODBC. Рекомендуемый способ установки unixODBC — использовать стандартные репозитории пакетов операционной системы Linux.

```
apt install unixodbc
```

2. Установите драйвер базы данных MariaDB unixODBC. Хотя у вас база данных MySQL, для обеспечения совместимости используется драйвер MariaDB unixODBC.

```
apt install odbc-mariadb
```

3. Проверьте расположение файлов конфигурации ODBC *odbcinst.ini* и *odbc.ini*.

```
odbcinst -j
```

Результат выполнения этой команды должен быть похож на следующий.

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

4. Чтобы настроить драйвер ODBC для мониторинга базы данных MySQL, вам нужно имя драйвера, которое находится в файле *odbcinst.ini*. В следующем примере файла *odbcinst.ini* имя драйвера — "MariaDB Unicode".

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

5. Скопируйте следующее содержимое в файл *odbc.ini* (замените <*password*> на пароль пользователя "zbx_monitor"). В этом руководстве в качестве адреса сервера MySQL используется "127.0.0.1" (localhost) для мониторинга сервера MySQL, установленного на той же машине, что и драйвер ODBC. Обратите внимание на имя источника данных (DSN) "test", которое потребуется при настройке веб-интерфейса Zabbix.

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

Настройте веб-интерфейс Zabbix и проверьте вашу конфигурацию

Чтобы настроить веб-интерфейс Zabbix, следуйте инструкциям в разделе *Настройка веб-интерфейса Zabbix* с учетом следующих изменений:

- В поле *Шаблоны* введите или выберите шаблон "MySQL by ODBC", который будет связан с узлом сети.
- Настройка *Интерфейсов* не требуется.
- Значение макроса {$MYSQL.DSN} в разделе *Унаследованные и макросы узла сети* в конфигурации узла сети MySQL server должно быть установлено в имя DSN из файла *odbc.ini*.

После настройки веб-интерфейса Zabbix вы можете просмотреть собранные метрики, настроить оповещения о проблемах и проверить вашу конфигурацию.

См. также

- Создание элемента данных - как начать мониторинг дополнительных метрик.

- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, а затем, если проблема не будет решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Мониторинг ODBC - как настроить ODBC в других дистрибутивах Linux и как начать мониторинг дополнительных метрик, связанных с базами данных, с помощью ODBC.
- Шаблон *MySQL by Zabbix agent* - дополнительная информация о шаблоне *MySQL by Zabbix agent*.
- Шаблон *MySQL by Zabbix agent 2* - дополнительная информация о шаблоне *MySQL by Zabbix agent 2*.
- Шаблон *MySQL by ODBC* - дополнительная информация о шаблоне *MySQL by ODBC*.

Мониторинг VMware с помощью Zabbix

Введение

На этой странице описаны шаги, необходимые для начала базового мониторинга VMware.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга VMware. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Мониторинг виртуальных машин или раздел Конфигурация руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС.

В зависимости от вашей среды некоторые шаги в этом руководстве могут незначительно отличаться. Это руководство основано на среде, работающей под управлением Ubuntu.

Предполагается, что VMware уже настроен. Это руководство не охватывает настройку VMware.

Настройка сервера Zabbix

Для мониторинга VMware необходимо включить процессы Zabbix *vmware collector*. Дополнительную информацию о выполнении мониторинга VMware см. в разделе Мониторинг виртуальных машин.

1. Откройте файл конфигурации сервера Zabbix.

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

2. Найдите параметр `StartVMwareCollectors` в файле конфигурации сервера Zabbix и установите его значение равным 2 или больше (значение по умолчанию — 0).

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

StartVMwareCollectors=2
```

3. Перезапустите сервер Zabbix.

```
systemctl restart zabbix-server
```

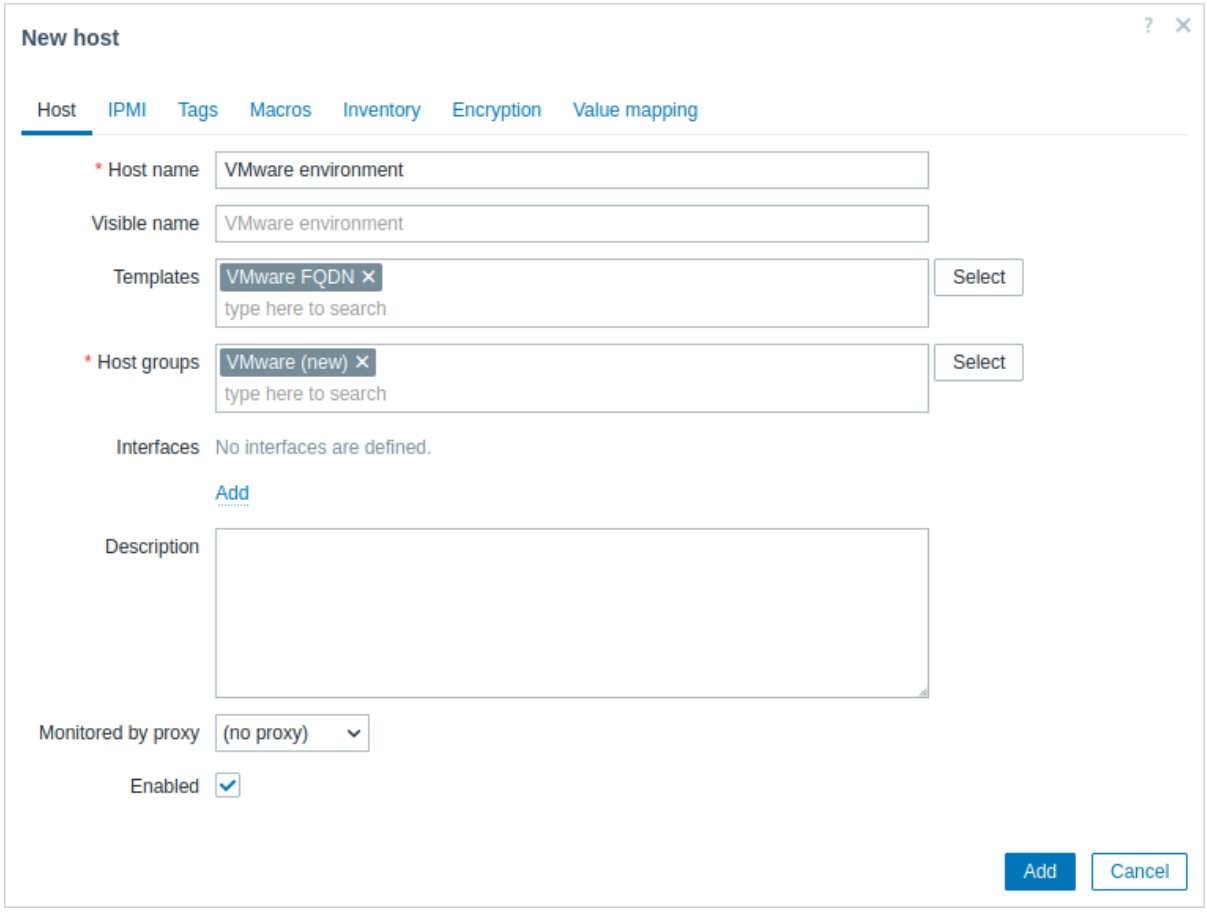
После запуска процессов *vmware collector* перейдите к следующему шагу.

Настройка веб-интерфейса Zabbix

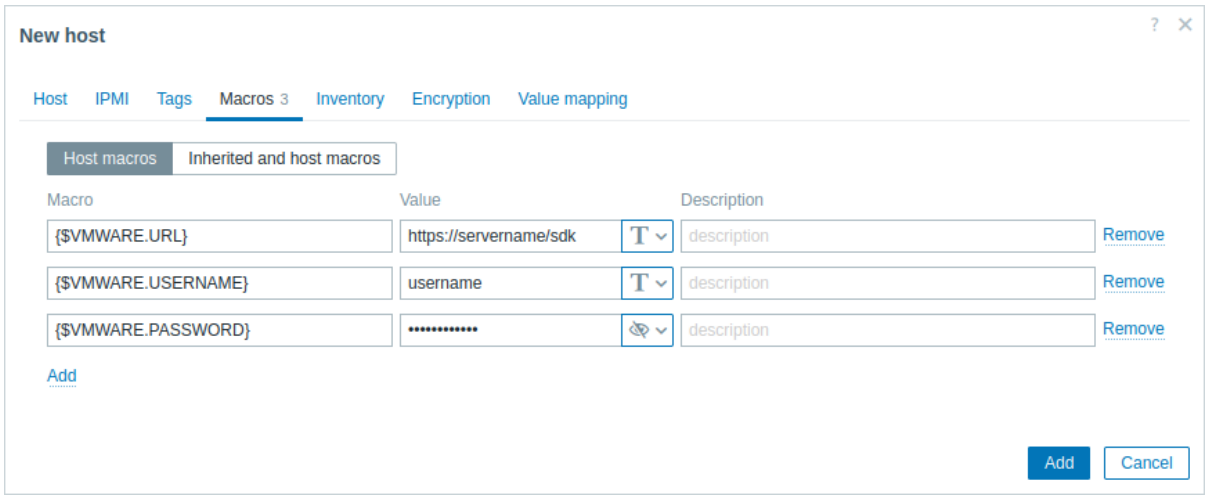
1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix:

- В поле *Host name* введите имя узла сети (например, ”VMware environment”).
- В поле *Templates* введите или выберите шаблон ”VMware FQDN” (или ”VMware”). Дополнительные сведения об этих шаблонах см. в разделе Мониторинг виртуальных машин.
- В поле *Host groups* введите или выберите группу узлов сети (например, новую группу узлов сети ”VMware”).



- На вкладке *Macros* задайте следующие макросы узла сети:
  - {$VMWARE.URL} - URL SDK службы VMware (vCenter или гипервизор ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - имя пользователя службы VMware
  - {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} службы VMware



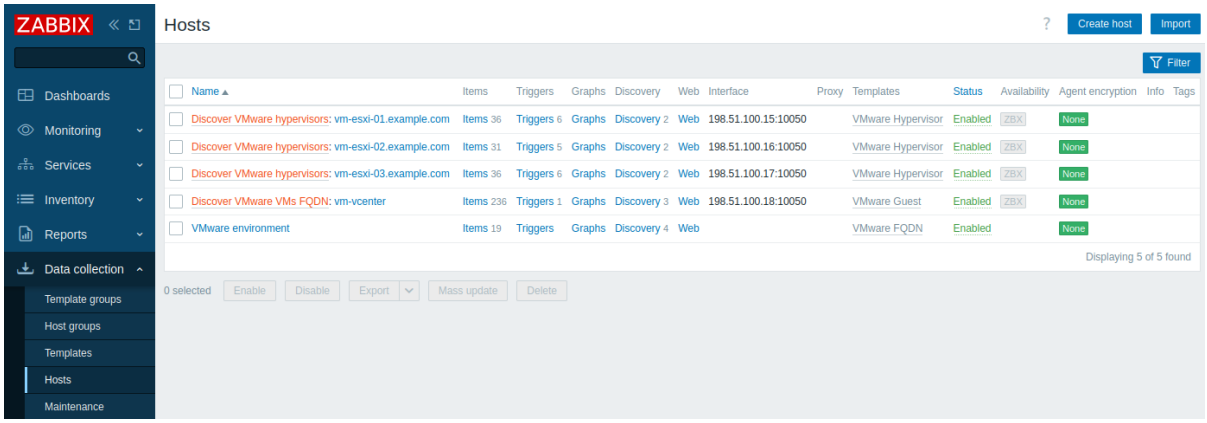
3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять вашу среду VMware.

Просмотр собранных метрик

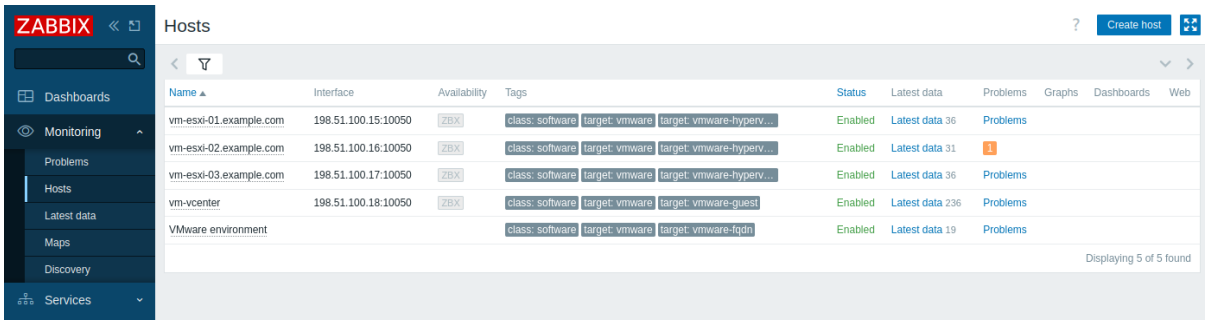
Поздравляем! На этом этапе Zabbix уже выполняет мониторинг вашей среды VMware.

В зависимости от конфигурации вашей среды VMware Zabbix может обнаружить и затем создать узлы сети для обнаруженных сущностей. Обратите внимание, что обнаружение и создание узлов сети также можно выполнить вручную, если это необходимо.

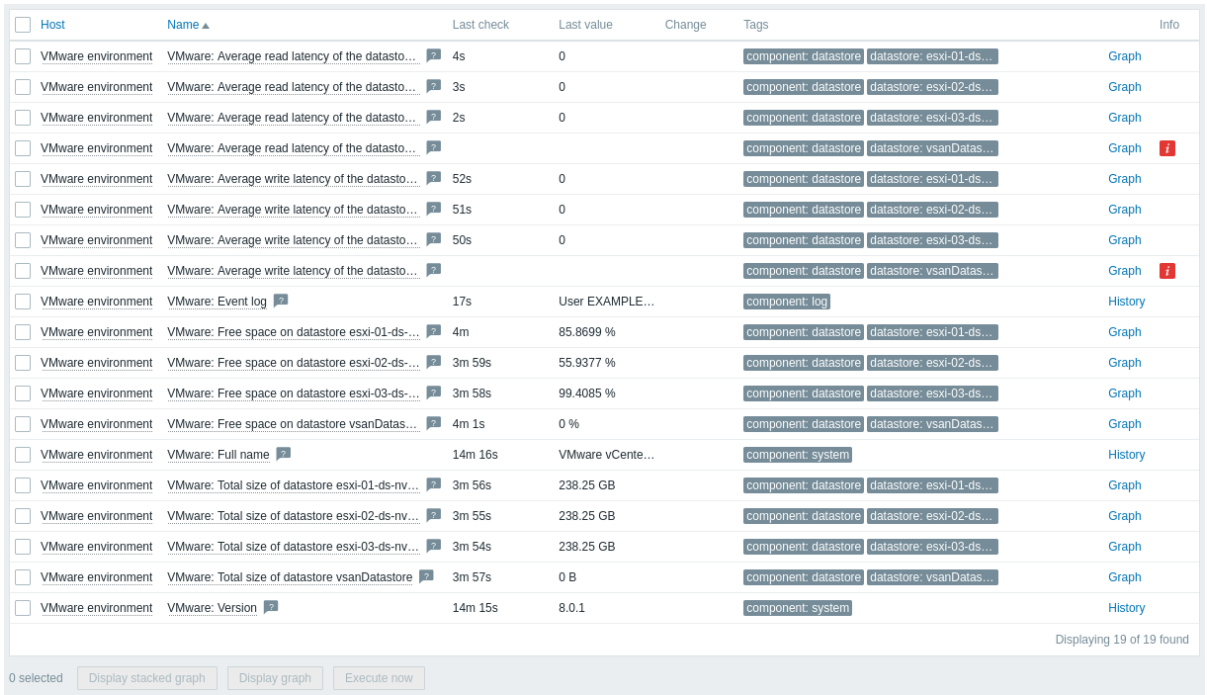
Чтобы просмотреть созданные узлы сети, перейдите в раздел меню *Сбор данных > Узлы сети*.



Чтобы просмотреть собранные метрики, перейдите в раздел меню *Мониторинг > Узлы сети* и нажмите *Последние данные* рядом с созданным узлом сети "VMware environment" или одним из узлов сети, созданных для обнаруженных сущностей.



Это действие откроет список всех последних метрик, собранных с выбранного узла сети.

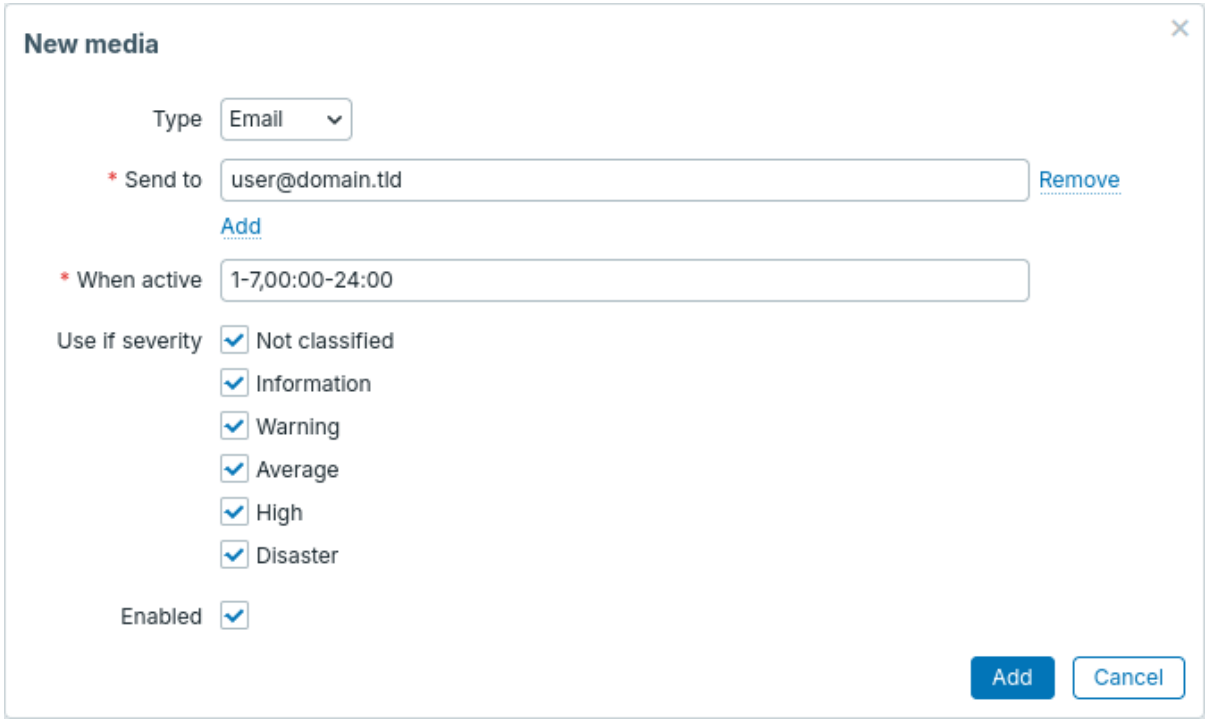


Обратите внимание, что у некоторых элементов данных нет данных и они находятся в состоянии *Не поддерживается*. Это связано с тем, что Zabbix не может найти допустимые счетчики производительности на конкретном хранилище данных, поскольку оно не включено в контролируемой среде VMware.

Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами. Это руководство содержит основные шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *User settings > Profile*, откройте вкладку *Media* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

См. также

- Создание элемента данных - как начать мониторинг дополнительных метрик.
- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, а затем, если проблема не будет решена в течение 45 минут, отправить сообщение менеджеру дата-центра).
- Мониторинг виртуальных машин - дополнительная информация о мониторинге VMware (процесс сбора данных, параметры конфигурации сервера, рекомендации по устранению неполадок и т. д.).
- Ключи элементов данных для мониторинга VMware - полный список метрик VMware, которые можно отслеживать с помощью Zabbix.
- Шаблон *VMware* - дополнительная информация о шаблоне *VMware*.
- Шаблон *VMware FQDN* - дополнительная информация о шаблоне *VMware FQDN*.

Мониторинг сетевого трафика с помощью Zabbix

Введение

На этой странице вы пройдете через шаги, необходимые для начала базового мониторинга сетевого трафика с помощью Zabbix.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга сетевого трафика. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, необходимо загрузить и установить сервер Zabbix, веб-интерфейс Zabbix и агент Zabbix в соответствии с инструкциями для вашей ОС.
Обратите внимание, что агент Zabbix следует установить на машине, для которой требуется мониторинг трафика.
Это может быть как тот же узел сети, где установлен сервер Zabbix, так и другой узел сети.

В этом руководстве будут приведены инструкции по настройке мониторинга сетевого трафика интерфейса *eth0* на отдельной машине с именем *Remote host*.

Настройка Zabbix для мониторинга

Агент Zabbix может собирать метрики в активном или пассивном режиме (одновременно). Дополнительную информацию см. в разделе Пассивные и активные проверки агента. В этом руководстве будет описан мониторинг с помощью пассивных проверок.

Настройка агента Zabbix

1. Откройте файл конфигурации агента на машине, где установлен агент:

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

2. Добавьте IP-адрес или DNS-имя вашего сервера Zabbix в параметр *Server*. Например:

```
Server=192.0.2.22
```

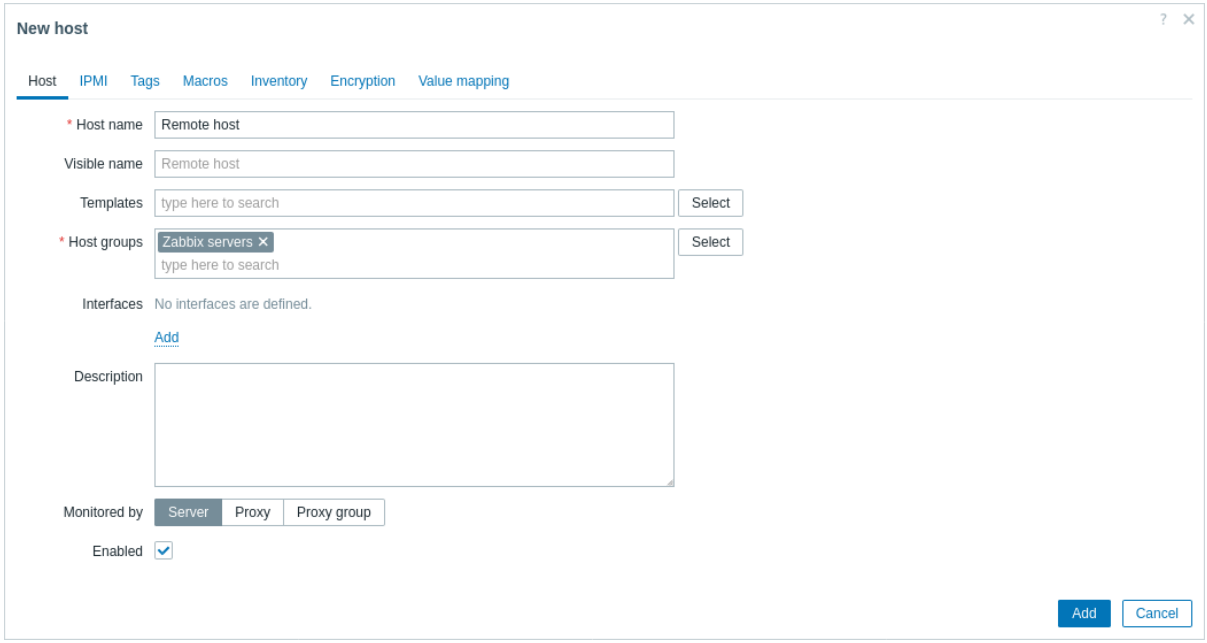
3. Перезапустите агент Zabbix:

```
systemctl restart zabbix-agent
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix, указав IP-адрес или DNS-имя машины, на которой установлен агент.

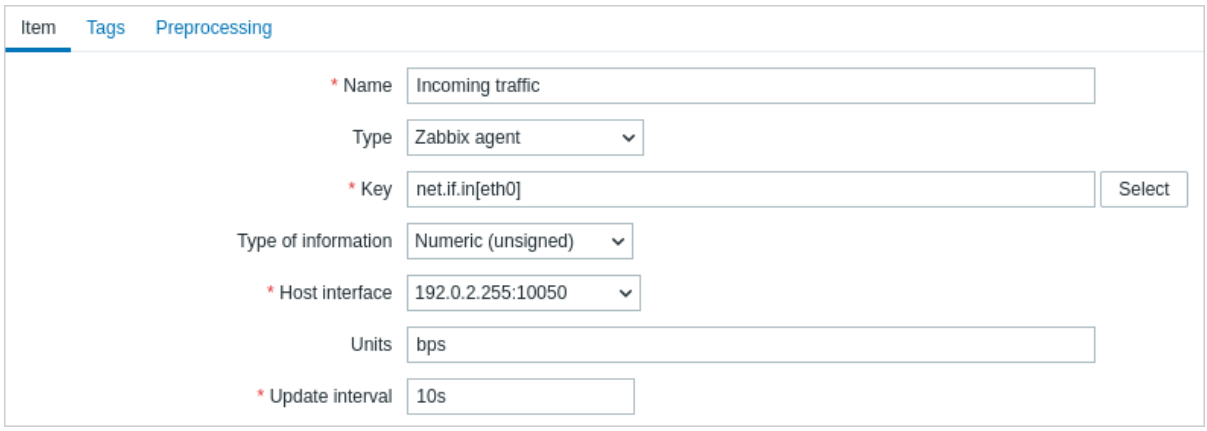


Создание элементов данных

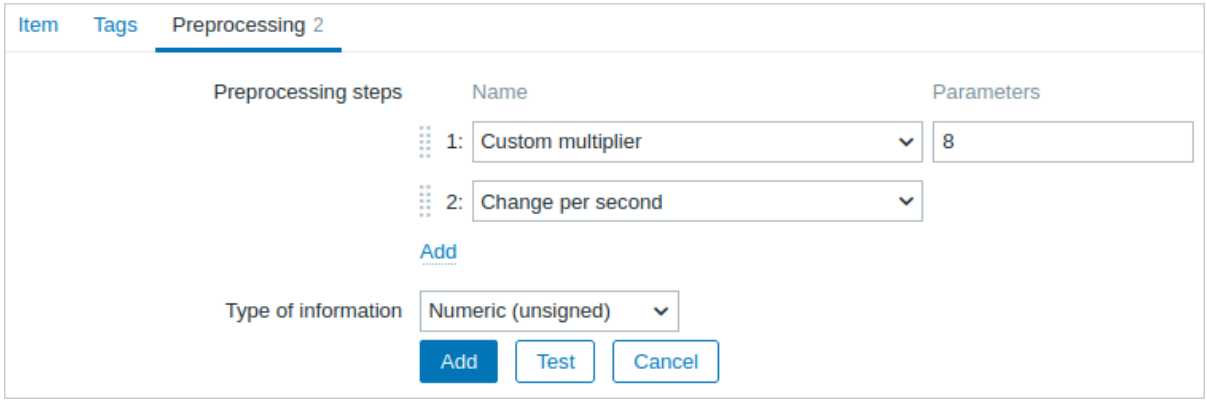
Следуйте инструкциям по созданию элемента данных, чтобы добавить элементы данных для мониторинга трафика, а именно:

- Входящий трафик
- Исходящий трафик
- Общий трафик

Простая настройка элемента данных для мониторинга входящего трафика будет выглядеть следующим образом:



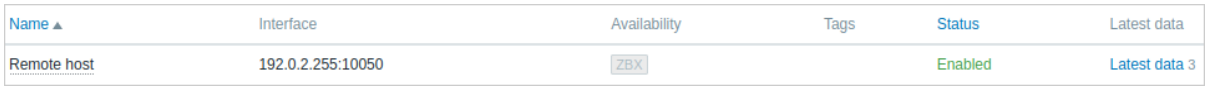
Чтобы собранные данные были пригодны для практического использования, при создании элементов данных можно задать некоторые шаги предобработки. В данном случае это может быть умножение на 8 (для преобразования байтов в биты) и представление в виде изменения за секунду.



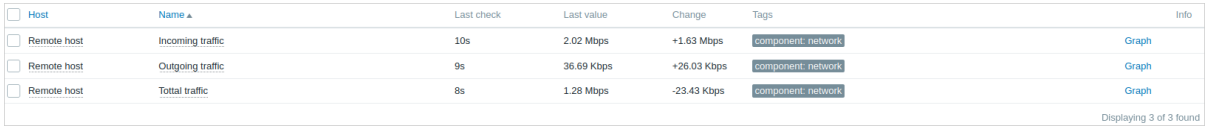
Просмотр собранных данных

Поздравляем! На этом этапе Zabbix уже отслеживает сетевой трафик.

Чтобы просмотреть собранные метрики, откройте раздел меню *Мониторинг > Узлы сети* и нажмите *Последние данные* в строке узла сети.

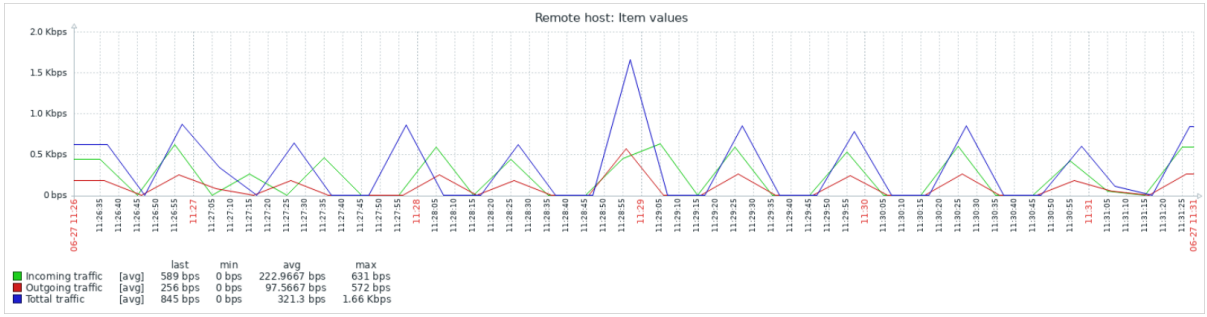


Вы увидите данные о трафике.



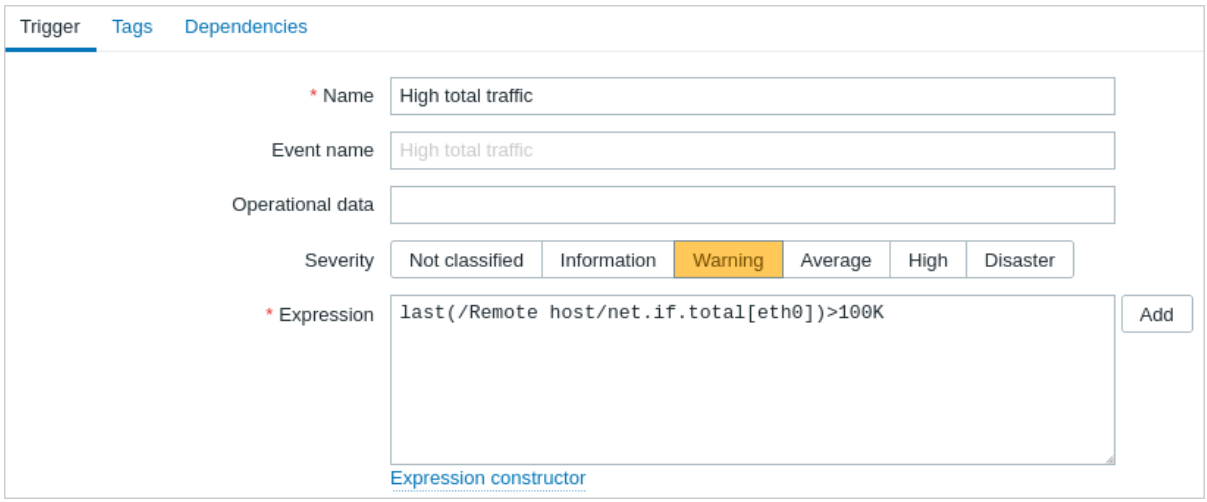
Просмотр графиков

Собранные данные можно отображать в виде графиков. Чтобы просмотреть их, в разделе *Последние данные* нажмите *График* в строке элемента данных или выберите нужные элементы данных и нажмите *Показать график* ниже.

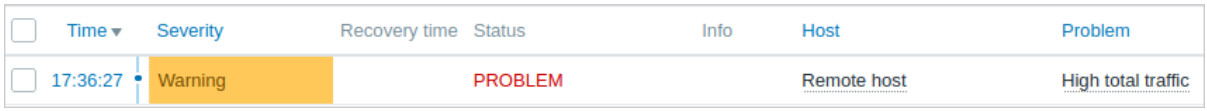


Настройка триггеров

Вы можете задать триггеры для обнаружения аномального сетевого трафика. См. инструкции по настройке триггера и добавьте триггер, сигнализирующий о том, что общий трафик слишком высок, например:



Теперь сделайте так, чтобы трафик превысил порог, заданный в выражении триггера, и перейдите в *Monitoring* > *Problems*, чтобы проверить, что проблема отображается там.



Настройка оповещений о проблемах

Существует несколько способов получать уведомления о проблеме. Электронная почта является самым популярным вариантом; следуйте инструкциям по настройке уведомления о проблеме по электронной почте. Вы также можете выбрать другие типы медиа, которые будут использоваться для доставки уведомлений.

См. также:

- Эскалации проблем — как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, а затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Подтверждение проблемы — как указать, что проблема известна, добавлять комментарии по ее устранению, подавлять или закрывать проблему.
- Мониторинг Linux с помощью агента Zabbix — как начать базовый мониторинг наиболее важных элементов данных, связав предварительно настроенный шаблон.

Мониторинг сетевого трафика с помощью активных проверок

Введение

На этой странице описаны шаги, необходимые для начала базового мониторинга сетевого трафика с помощью Zabbix с использованием активных проверок.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга сетевого трафика с использованием активных проверок. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо загрузить и установить сервер Zabbix, веб-интерфейс Zabbix и агент Zabbix в соответствии с инструкциями для вашей ОС.
Обратите внимание, что агент Zabbix следует установить на машине, для которой требуется мониторинг трафика.
Это может быть как тот же узел сети, где установлен сервер Zabbix, так и другой узел сети.

В этом руководстве будут приведены инструкции по настройке мониторинга сетевого трафика интерфейса *eth0* на отдельной машине с именем *Remote host*.

Настройка Zabbix для мониторинга

Агент Zabbix может собирать метрики в активном или пассивном режиме (одновременно). Дополнительную информацию см. в разделе Пассивные и активные проверки агента. В этом руководстве будет описан мониторинг с помощью **активных проверок**.

Настройка агента Zabbix

1. Откройте файл конфигурации агента на машине, где установлен агент.

Если вы используете агент Zabbix:

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

Если вы используете агент Zabbix 2:

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

2. Добавьте IP-адрес или DNS-имя (и необязательный порт) вашего сервера Zabbix в параметр ServerActive. Например:

```
ServerActive=192.0.2.1:10051
```

Агент Zabbix будет использовать этот адрес для подключения к порту трапперов сервера Zabbix (по умолчанию: 10051) и запроса данных конфигурации для активных проверок.

3. Определите параметр Hostname, который должен совпадать с именем узла сети, заданным в веб-интерфейсе Zabbix. В нашем примере это:

```
Hostname=Remote host
```

Значение Hostname должно совпадать, поскольку для активных проверок агент Zabbix использует его для получения правильной конфигурации узла сети с сервера. В частности, агент инициирует подключение к серверу и идентифицирует себя с помощью значения Hostname. Затем сервер предоставляет конфигурацию мониторинга для этого узла сети. Если эти значения различаются, агент не получит соответствующую конфигурацию, что приведет к отсутствию метрик или проблемам с мониторингом.

4. Перезапустите агент Zabbix.

Если вы используете агент Zabbix:

```
systemctl restart zabbix-agent
```

Если вы используете агент Zabbix 2:

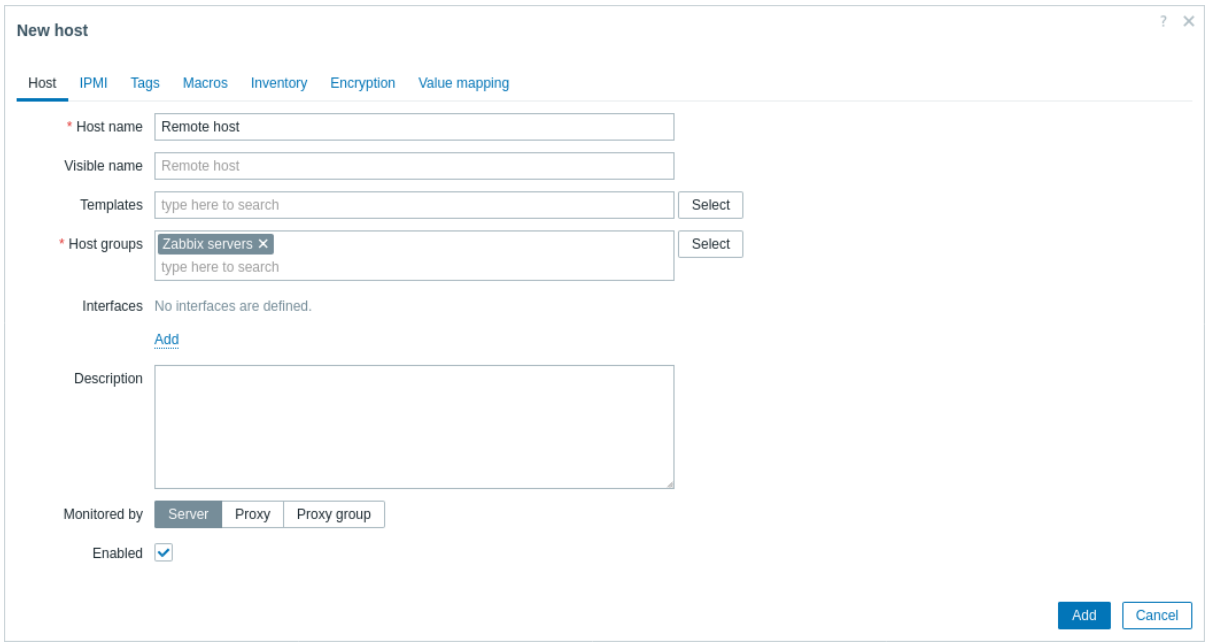
```
systemctl restart zabbix-agent2
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

- В поле *Host name* введите имя узла сети (например, ”Remote host”), которое соответствует значению параметра Hostname, ранее заданному в файле конфигурации агента.
- В поле *Host groups* введите или выберите группу узлов сети (например, ”Zabbix servers”).
- Оставьте *Interfaces* не заданным; интерфейс не нужен для активных проверок, поскольку агент инициирует соединение с сервером вместо того, чтобы ожидать соединения от сервера.



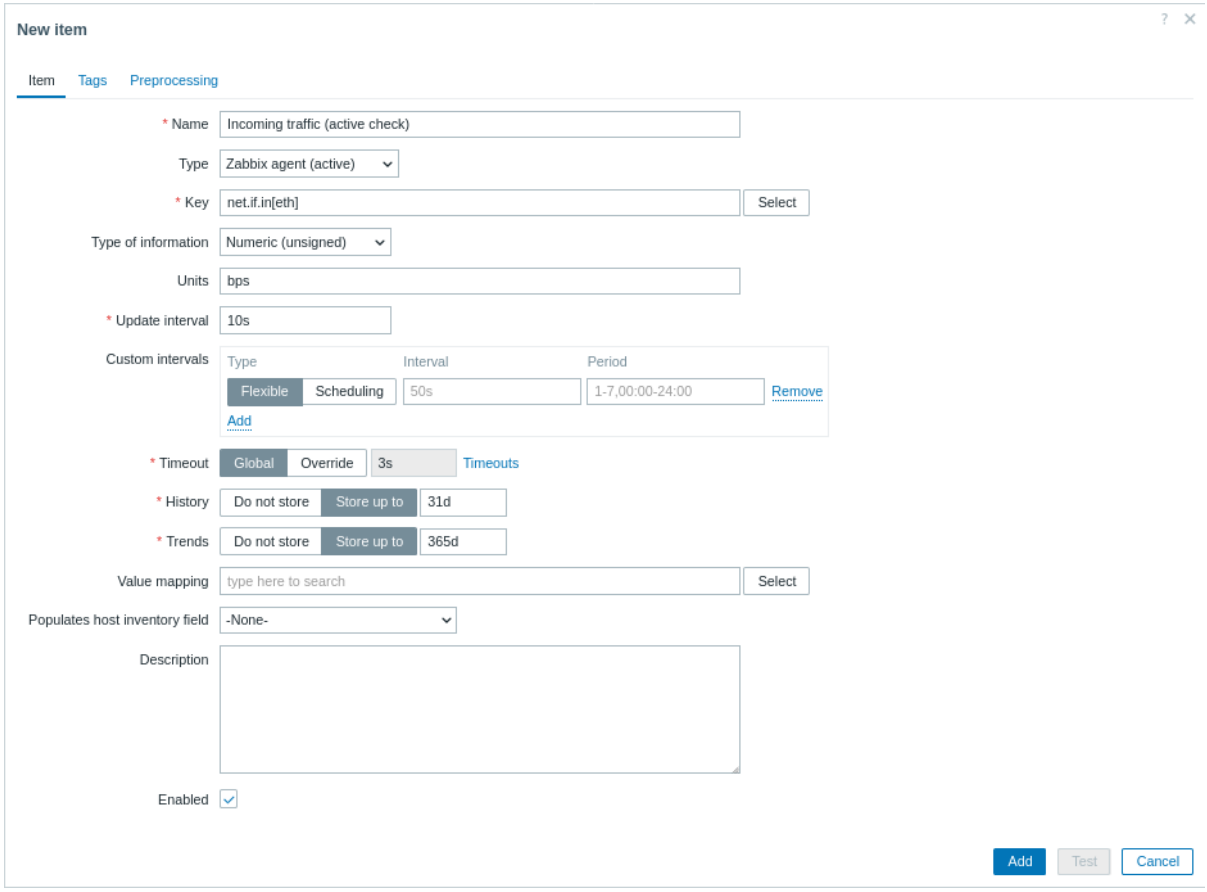
3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять контролируемую машину Linux.

Создание элементов данных

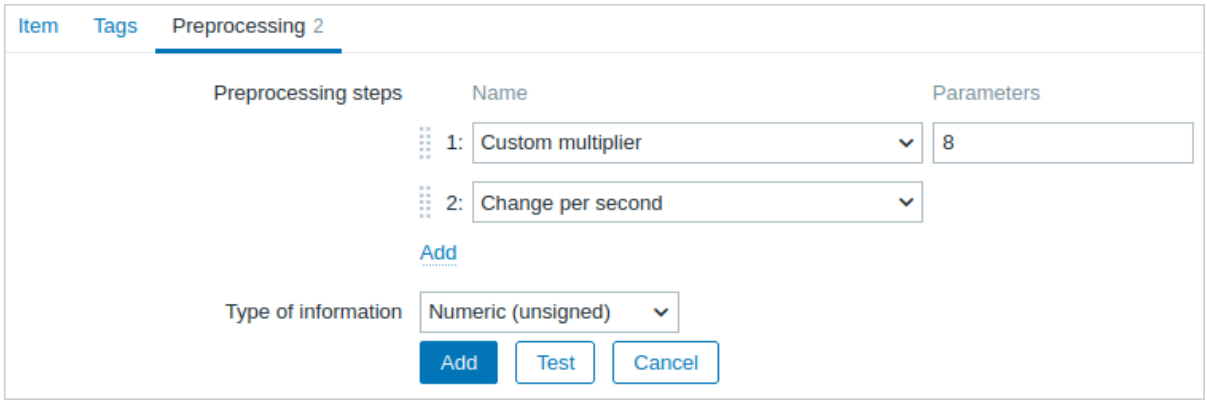
Следуйте инструкциям по созданию элемента данных, чтобы добавить элементы данных для мониторинга трафика, а именно:

- Входящий трафик
- Исходящий трафик
- Общий трафик

Простая настройка элемента данных для мониторинга входящего трафика с активной проверкой будет выглядеть следующим образом:



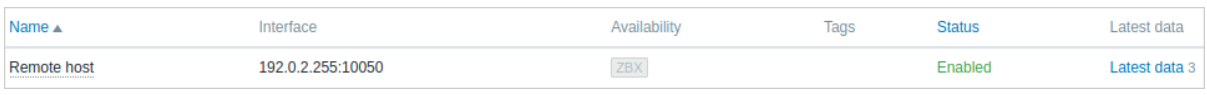
Чтобы сделать собранные данные пригодными для практического использования, при создании элементов данных можно задать некоторые шаги предобработки. В данном случае это может быть умножение на 8 (для преобразования байтов в биты) и представление в виде изменения в секунду.



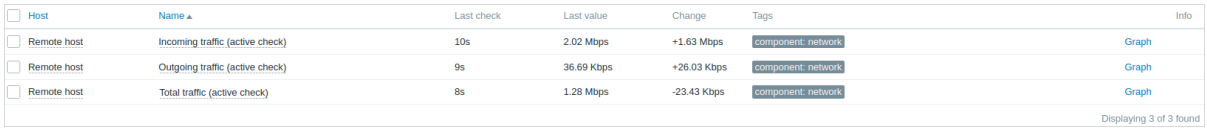
Просмотр собранных данных

Поздравляем! На этом этапе Zabbix уже отслеживает сетевой трафик в вашей сети.

Чтобы просмотреть собранные метрики, откройте раздел меню *Monitoring > Hosts* и нажмите *Latest data* в строке узла сети.



Вы увидите данные о трафике.



См. также:

- Просмотр графиков - как отображать собранные данные в виде графиков.
- Настройка триггеров - как настроить триггеры для обнаружения аномального сетевого трафика.

- Настройка оповещений о проблемах - как настроить уведомления о проблемных ситуациях.
- Эскалации проблем - как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру дата-центра).
- Подтверждение проблемы - как указать, что проблема известна, добавлять комментарии по ее устранению, подавлять или закрывать проблему.
- Мониторинг Linux с помощью агента Zabbix - как начать базовый мониторинг важнейших элементов данных, связав предварительно настроенный шаблон.
- Установка из пакетов - как устанавливать компоненты Zabbix с использованием официальных пакетов RPM и DEB для различных дистрибутивов Linux, обеспечивая доступ к новейшим функциям и исправлениям ошибок.

Мониторинг веб-сайтов с помощью элементов данных Browser

Введение

На этой странице пошагово описаны действия, необходимые для начала базового мониторинга веб-сайтов с помощью элементов данных Browser.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга веб-сайтов с помощью элементов данных Browser. Если вам нужны возможности глубокой настройки или более продвинутая конфигурация, см. страницу Browser items или раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС.

В зависимости от вашей среды некоторые шаги в этом руководстве могут незначительно отличаться. Это руководство основано на среде, работающей под Ubuntu.

Настройка WebDriver

Элементы данных браузера требуют фреймворк автоматизации (либо Selenium Server, либо обычный WebDriver, например ChromeDriver) в качестве конечной точки веб-тестирования, которая управляет браузером и взаимодействует с ним, выполняя тестовые команды, такие как нажатие кнопок или ввод текста. В качестве примера в этом руководстве будет использоваться Selenium Server с Chrome в контейнере Docker.

Предполагается, что Docker уже настроен. В этом руководстве не рассматривается настройка Docker. Инструкции по установке см. в разделе Install Docker Engine on Ubuntu.

1. Запустите Selenium Server с Chrome в контейнере Docker со следующими параметрами:

- **docker run --name browser** - запускает новый контейнер Docker с именем "browser";
- **-p 4444:4444** - сопоставляет порт 4444 на вашем хосте с портом 4444 в контейнере (это порт, который Selenium Server использует для приема команд);
- **-p 7900:7900** - сопоставляет порт 7900 на вашем хосте с портом 7900 в контейнере (это порт, который использует сервер Virtual Network Computing (VNC), позволяя удаленно просматривать графический интерфейс браузера; требуется VNC-клиент);
- **--shm-size="2g"** - выделяет контейнеру 2 ГБ общей памяти (это важно для корректной работы Chrome, так как ему может требоваться значительный объем общей памяти, чтобы избежать сбоев);
- **-d** - запускает контейнер в отсоединенном режиме, то есть он будет работать в фоновом режиме;
- **selenium/standalone-chrome:latest** - указывает используемый образ Docker; в данном случае это последняя версия Selenium Server with Chrome.

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

2. Убедитесь, что контейнер Docker `browser` запущен и доступен.

- Получите IP-адрес контейнера (в этом примере, 192.0.2.1):

```
ip addr

##### 1: lo: <LOOPBACK,UP,LOWER_UP>
```

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

- Проверьте соединение с контейнером с помощью Ncat:

```
nc -zv 192.0.2.1 4444

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

- Получите содержимое веб-страницы из Selenium Server с помощью curl:

```
curl -L 192.0.2.1:4444

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

Note:
Для устранения неполадок см. Docker documentation.

Настройка сервера Zabbix

Элементы данных браузера выполняются и обрабатываются процессами Zabbix типа *browser poller*. Их необходимо включить, изменив параметр конфигурации сервера StartBrowserPollers. Кроме того, параметр WebDriverURL должен указывать ранее настроенную конечную точку веб-тестирования.

По умолчанию параметр StartBrowserPollers имеет значение 1, поэтому необходимо указать только конечную точку веб-тестирования.

1. Откройте файл конфигурации сервера Zabbix:

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

2. Найдите и задайте параметр WebDriverURL в файле конфигурации сервера Zabbix:

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

WebDriverURL=192.0.2.1:4444
```

3. Перезапустите сервер Zabbix.

```
systemctl restart zabbix-server
```

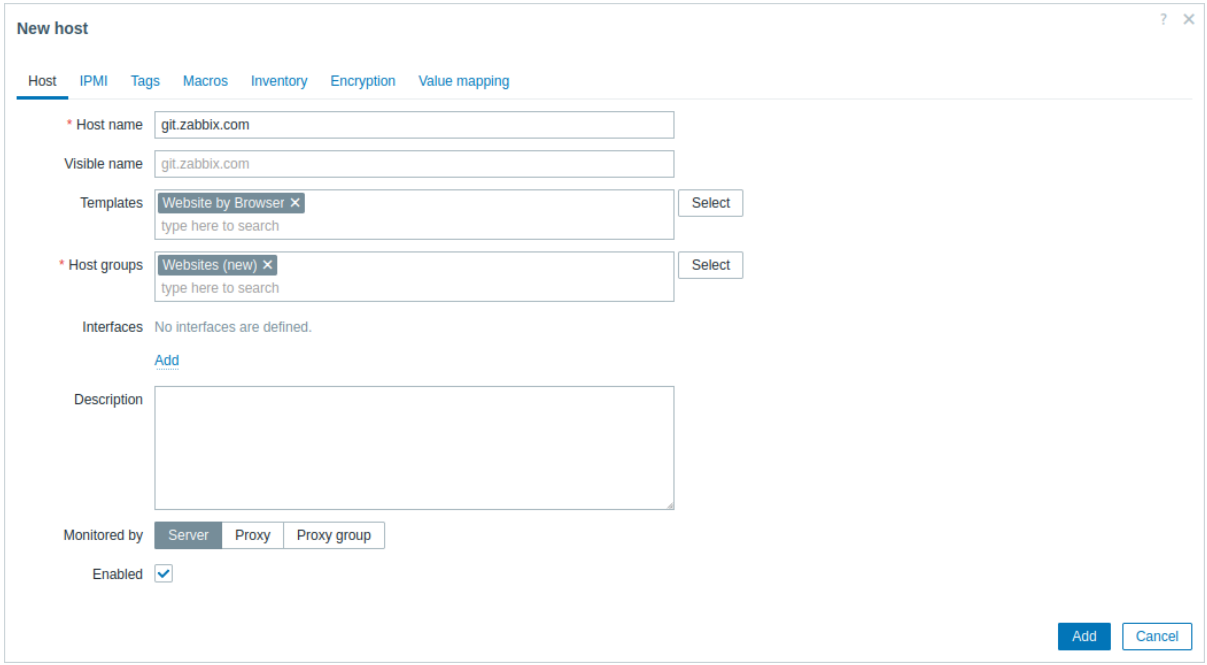
Настройка веб-интерфейса Zabbix

1. Войдите в веб-интерфейс Zabbix.

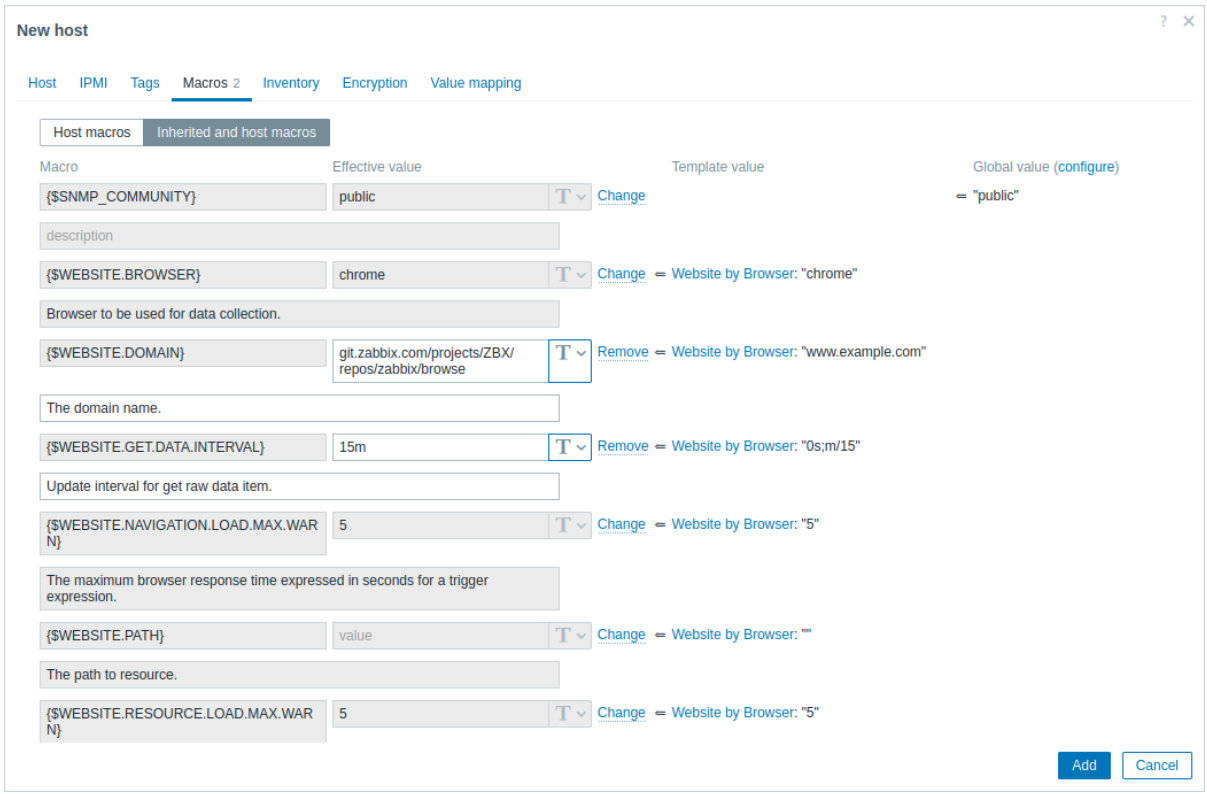
2. Создайте узел сети в веб-интерфейсе Zabbix:

- В поле *Host name* введите имя узла сети (например, ”git.zabbix.com”).
- В поле *Templates* введите или выберите шаблон ”Website by Browser”. Дополнительные сведения об этом шаблоне см. в разделе Website by Browser.

- В поле *Host groups* введите или выберите группу узлов сети (например, новую группу узлов сети ”Websites”).



- На вкладке *Macros* переключитесь на *Inherited and host macros*, найдите следующие макросы и нажмите *Change* рядом со значением макроса, чтобы обновить его:
  - {$WEBSITE.DOMAIN} - Доменное имя (например, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Интервал обновления данных элемента данных (например, 15m)

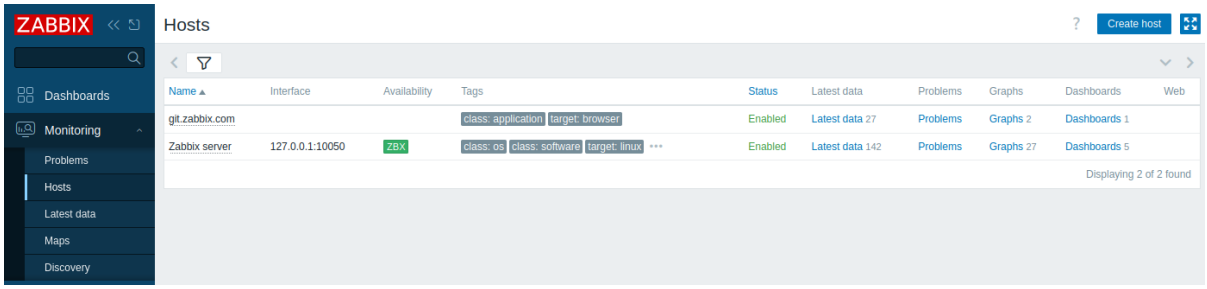


3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять веб-сайт, который вы хотите мониторить.

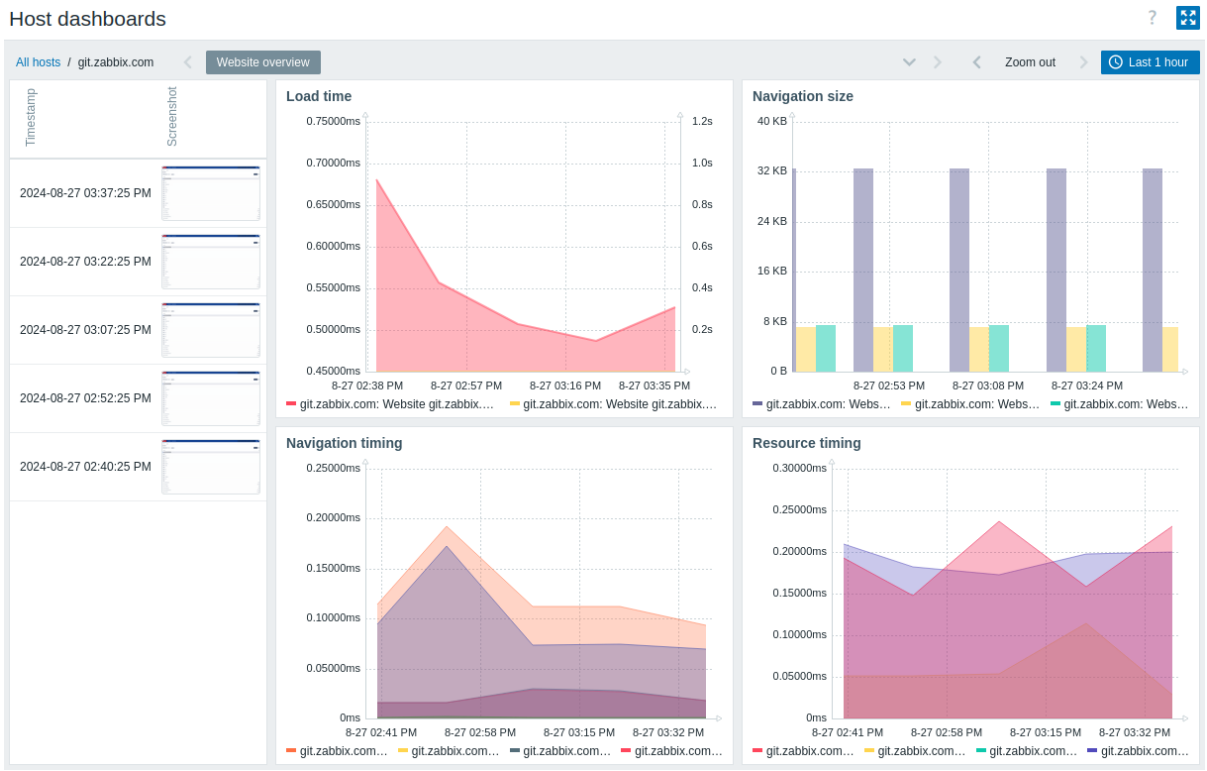
Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает указанный вами веб-сайт.

Чтобы просмотреть собранные метрики, перейдите в раздел меню *Monitoring > Hosts* и нажмите *Dashboards* рядом с узлом сети.



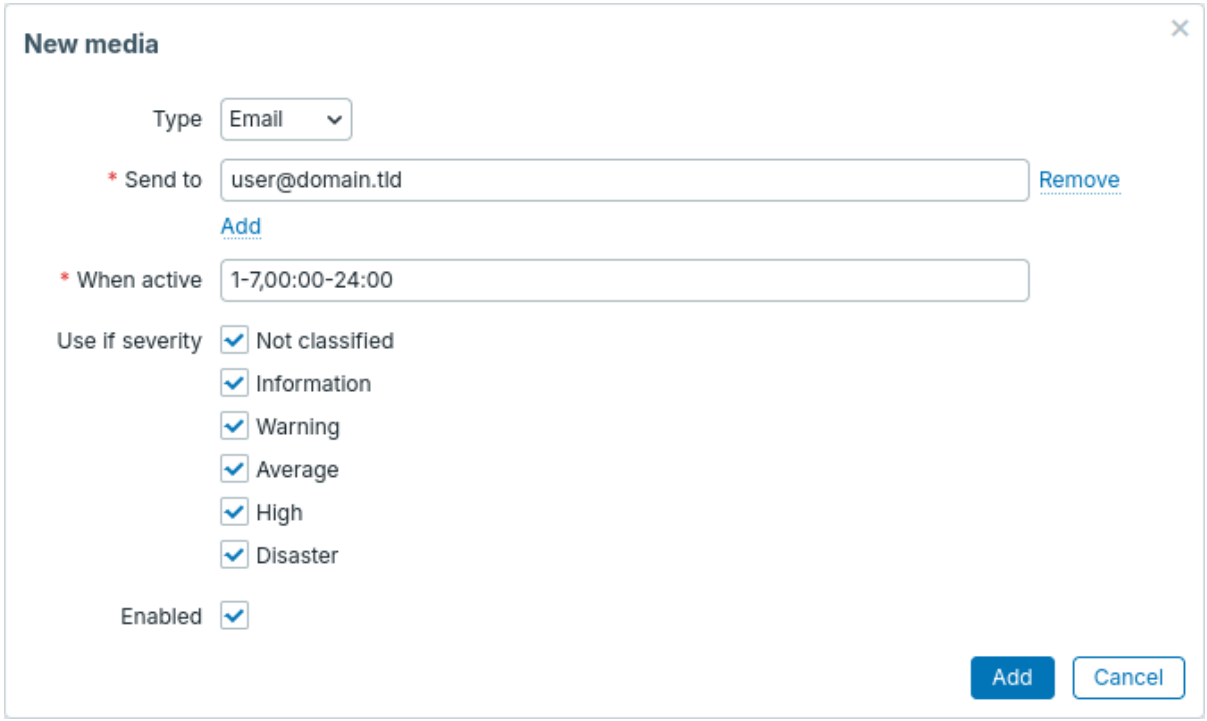
Это действие откроет панель узла сети (настроенную на уровне шаблона) с наиболее важными метриками, собранными с веб-сайта.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами. Это руководство содержит основные шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя* > *Профиль*, откройте вкладку *Media* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию

Чтобы проверить вашу конфигурацию, мы можем смоделировать реальную проблему, обновив конфигурацию узла сети в веб-интерфейсе Zabbix.

1. Откройте конфигурацию узла сети вашего веб-сайта в Zabbix.

2. Перейдите на вкладку *Macros* и выберите *Inherited and host macros*.

3. Нажмите *Change* рядом, например, со ранее настроенным значением макроса {$WEBSITE.DOMAIN} и задайте неверное доменное имя (например, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Нажмите *Update*, чтобы обновить конфигурацию узла сети.

5. Через несколько мгновений Zabbix обнаружит проблему "Failed to get JSON of the requested website", поскольку не сможет подключиться к указанному веб-сайту. Проблема появится в *Monitoring > Problems*.



Если настроены оповещения, вы также получите уведомление о проблеме.

6. Верните значение макроса к предыдущему значению, чтобы устранить проблему и продолжить мониторинг веб-сайта.

См. также

- Создание элемента данных - как начать мониторинг дополнительных метрик.
- Эскалации проблем - как создать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Элементы данных Browser - как настроить элементы данных Browser.
- Шаблон Website by Browser - дополнительная информация о шаблоне *Website by Browser*.

Мониторинг сертификатов веб-сайтов с помощью Zabbix agent 2 (пассивный)

Введение

Это руководство предоставляет подробный обзор того, как настроить и отслеживать сертификаты SSL/TLS с помощью ключа `web.certificate.get` в агенте 2 Zabbix. Оно предназначено для упрощения мониторинга сертификатов для одного или нескольких веб-сайтов, позволяя администраторам быстро выявлять потенциальные проблемы, такие как просроченные или недействительные сертификаты.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга сертификатов веб-сайтов. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо загрузить и установить сервер Zabbix, веб-интерфейс Zabbix и агент Zabbix 2 в соответствии с инструкциями для вашей ОС. В этом руководстве предполагается, что сервер Zabbix и агент установлены на одной машине; поэтому в конфигурации используется `127.0.0.1`.

Настройка агента Zabbix 2

1. Откройте файл конфигурации агента:

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

2. Установите для параметра Server значение 127.0.0.1, поскольку агент и сервер работают на одном компьютере:

```
Server=127.0.0.1
```

3. Сохраните файл и перезапустите службу агента Zabbix 2:

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

4. После установки и настройки агента Zabbix 2 проверьте его доступность с помощью команды:

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

Агент Zabbix 2 по умолчанию включает плагин WebCertificate, поэтому отдельная установка или настройка не требуется.

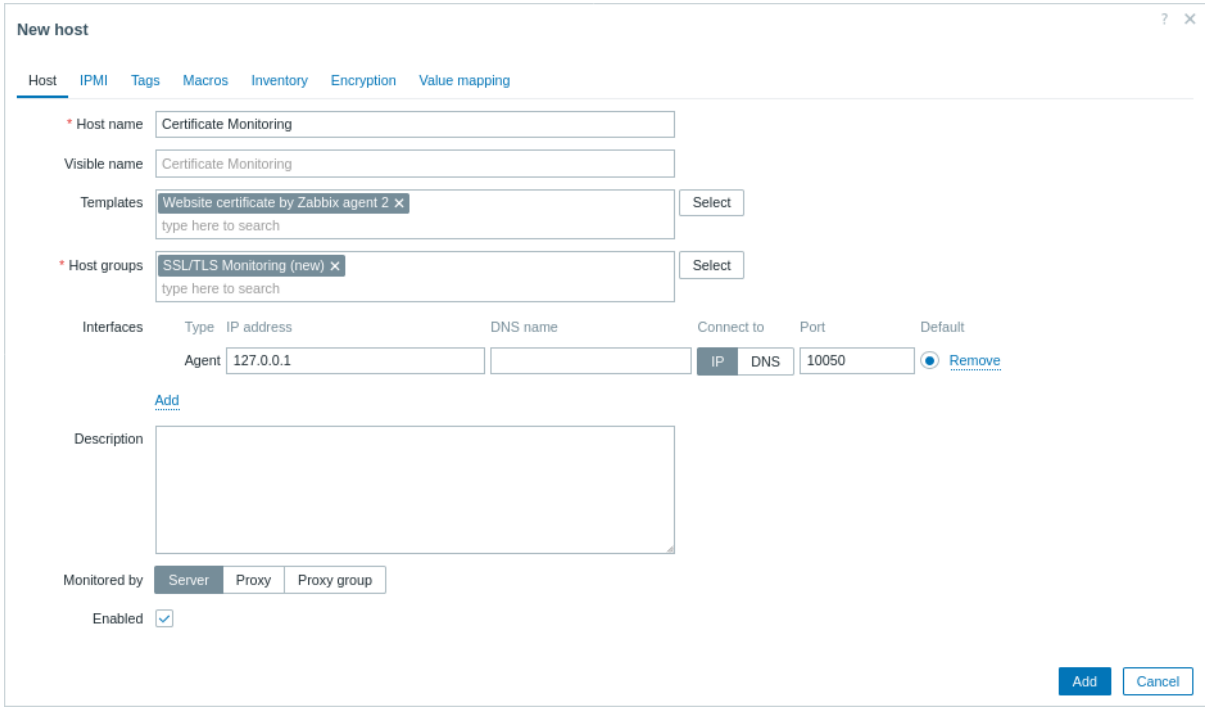
Настройка веб-интерфейса Zabbix

1. Войдите в веб-интерфейс Zabbix.

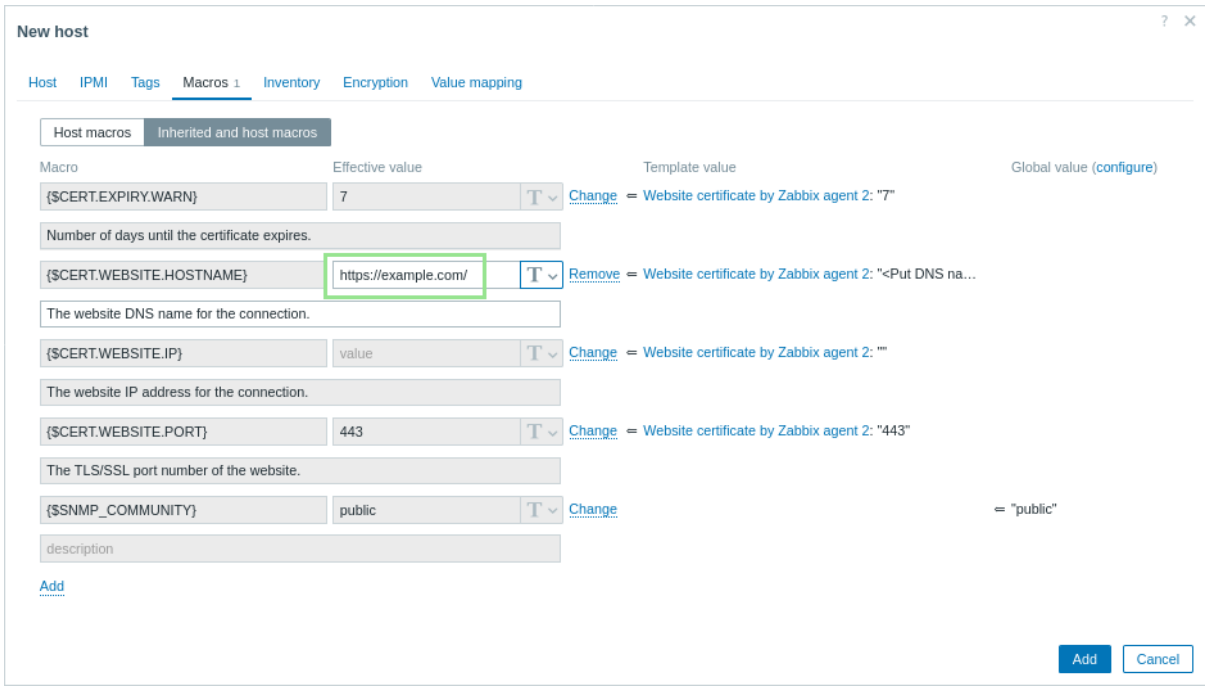
2. Перейдите в *Мониторинг* > *Узлы сети*.

3. Нажмите на существующий узел сети, для которого вы хотите отслеживать сертификаты веб-сайтов, или создайте узел сети, если это необходимо:

- В поле *Имя узла сети* введите имя узла сети (например, ”Certificate Monitoring”).
- В поле *Шаблоны* введите или выберите шаблон ”Website certificate by Zabbix agent 2”, который будет привязан к узлу сети.
- В поле *Группы узлов сети* введите или выберите группу узлов сети (например, ”SSL/TLS Monitoring”).
- В поле *Интерфейсы* добавьте интерфейс типа ”Agent” и укажите IP-адрес. В этом примере используется ”127.0.0.1”.



- На вкладке *Макросы* переключитесь на *Наследуемые и макросы узла сети*, найдите следующие макросы и нажмите *Изменить* рядом со значением макроса, чтобы обновить его:
  - {$CERT.WEBSITE.HOSTNAME} - введите в качестве значения нужное DNS-имя веб-сайта.



4. Нажмите *Добавить*, чтобы добавить узел сети.

Чтобы отслеживать несколько веб-сайтов, вы можете указать в макросе {$CERT.WEBSITE.HOSTNAME} список их имен хостов через запятую. При необходимости вы также можете указать в макросе {$CERT.WEBSITE.PORT} список портов через запятую, где каждый порт соответствует соответствующему имени хоста по порядку. Например, чтобы отслеживать example.com и example.org на порту 8443:

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

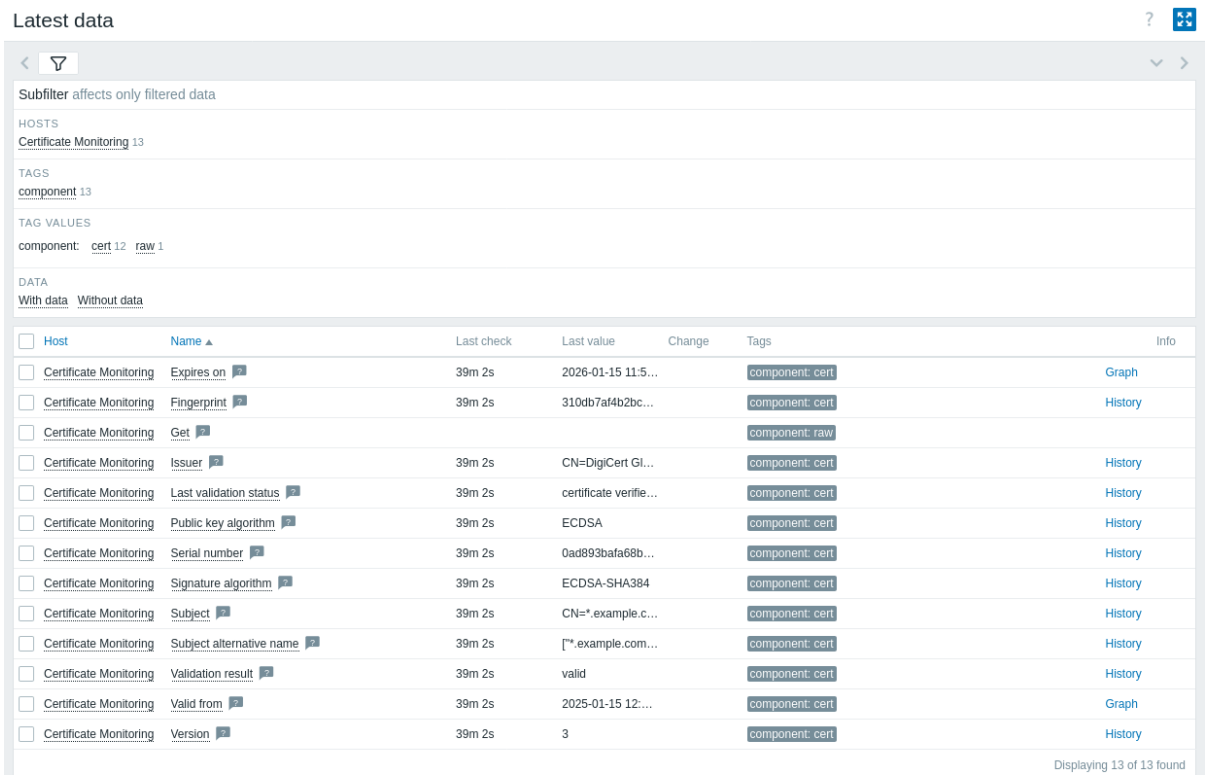
Для каждого веб-сайта, указанного в макросе {$CERT.WEBSITE.HOSTNAME}, Zabbix создаст соответствующий набор элементов данных и триггеров. Это позволяет выполнять отдельный мониторинг и оповещение для SSL-сертификата каждого веб-сайта.

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

Просмотр собранных метрик

Поздравляем! На этом этапе Zabbix уже отслеживает нужный веб-сертификат.

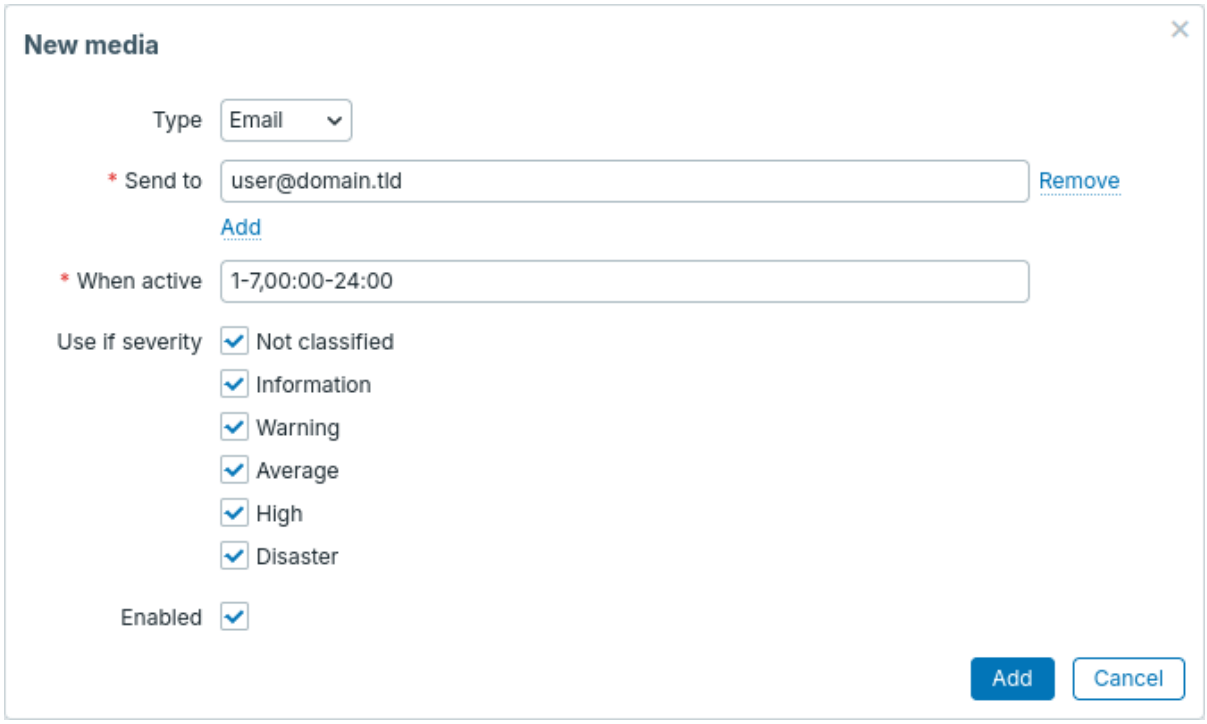
Чтобы просмотреть собранные метрики, перейдите в раздел меню *Monitoring > Hosts* и нажмите *Latest data* рядом с узлом сети, чтобы просмотреть все последние собранные метрики в виде списка, например дату истечения срока действия, издателя и субъект.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в инфраструктуре. Это руководство содержит базовые шаги настройки для отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя > Профиль*, откройте вкладку *Media* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию

Чтобы проверить вашу конфигурацию, мы можем смоделировать реальную проблему, обновив конфигурацию узла сети в веб-интерфейсе Zabbix.

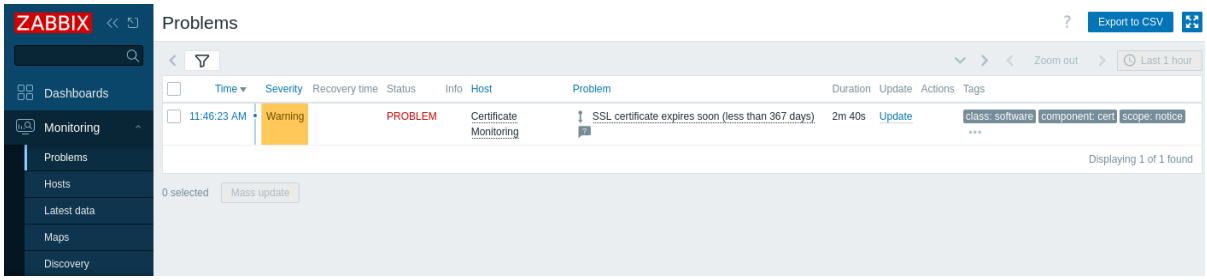
1. Откройте конфигурацию узла сети ”Certificate Monitoring” в Zabbix.

2. Перейдите на вкладку *Macros* и выберите *Inherited and host macros*.

3. Нажмите *Change* рядом с ранее настроенным значением макроса {$CERT.EXPIRY.WARN} и задайте очень большое число дней (достаточно значения больше 365 дней), чтобы получать предупреждение до истечения срока действия сертификата.

4. Нажмите *Update*, чтобы обновить конфигурацию узла сети.

5. Через несколько мгновений Zabbix обнаружит проблему "SSL certificate expires soon" с указанием количества дней до истечения срока действия. Проблема появится в *Monitoring > Problems*.



Если настроены оповещения, вы также получите уведомление о проблеме.

6. Измените значение макроса обратно на прежнее, чтобы устранить проблему и продолжить мониторинг значений сертификата.

См. также

- Zabbix agent 2 - перечисляет ключи элементов данных.
- Шаблон *Website certificate by Zabbix agent 2* - дополнительная информация о шаблоне *Website certificate by Zabbix agent 2*.
- Шаблон *Website certificate by Zabbix agent 2 active* - дополнительная информация о шаблоне *Website certificate by Zabbix agent 2 active*.
- Monitor websites with Browser items - как начать базовый мониторинг веб-сайтов с помощью элементов данных Browser.

Мониторинг сетевого коммутатора или маршрутизатора с помощью Zabbix

Введение

Это руководство проведет вас через шаги, необходимые для начала базового мониторинга вашего сетевого коммутатора или маршрутизатора с помощью Zabbix. В качестве примера используется маршрутизатор Cisco, но процедура применима к любому сетевому устройству с поддержкой SNMP.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и сетевых администраторов, которые хотят быстро включить базовый мониторинг сетевых устройств. Если вам требуется глубокая настройка или расширенные параметры конфигурации, обратитесь к странице SNMP agent или разделу Configuration руководства Zabbix.

Предварительные требования

Перед тем как продолжить, убедитесь, что у вас есть:

- Установлены сервер Zabbix и веб-интерфейс Zabbix: установите их согласно инструкциям для вашей операционной системы (см. Установка из пакетов и Установка веб-интерфейса).
- Установлен агент Zabbix, если выполняется мониторинг локальных сетевых метрик.
- Устройство с поддержкой SNMP: сетевой коммутатор или маршрутизатор (например, маршрутизатор Cisco) с включенным SNMP.
- Установлены MIB files: установка MIB-файлов позволяет Zabbix преобразовывать числовые OID в понятные человеку имена и описания. Без надлежащей поддержки MIB вы можете видеть только числовые значения, что затрудняет настройку элементов данных и устранение неполадок.

Чтобы установить MIB-файлы в Ubuntu:

1. Установите пакет загрузчика MIB:

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

Если вам нужно добавить MIB-файлы, специфичные для производителя (например, от Cisco, Juniper), поместите их в соответствующий каталог MIB:

- Для систем на базе Linux распространенные расположения включают `/usr/share/snmp/mibs/` или `/usr/local/share/snmp`
- Для установок Zabbix MIB-файлы можно хранить в `/var/lib/zabbix/mibs/`.

Убедитесь, что переменная окружения `MIBDIRS` или файл `snmp.conf` содержит правильный путь.

Чтобы проверить, что система распознает новые MIB, используйте:

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

Подробные инструкции см. в документации к вашей SNMP-библиотеке:

- Cisco MIBs
- Juniper MIBs

2. Отредактируйте `/etc/snmp/snmp.conf` и закомментируйте строку, начинающуюся с `mibs :`, чтобы разрешить системе загружать все доступные MIB.

3. Проверьте работу, выполнив `snmpwalk` (например, `snmpwalk -v 2c -c <your_community_string> <device_IP>`) и убедитесь, что OID отображаются с описательными именами.

В зависимости от вашей среды некоторые шаги в этом руководстве могут немного отличаться. Это руководство основано на среде с Ubuntu и мониторингом сетевого устройства Cisco Catalyst 3750V2-24FS.

Предполагается, что ваше сетевое устройство уже физически установлено и подключено.

Настройка сетевого устройства (пример маршрутизатора Cisco)

Для мониторинга через SNMP необходимо настроить сетевое устройство так, чтобы оно разрешало SNMP-запросы. Приведенный ниже пример относится к SNMPv2 и не учитывает существующие настройки. Внимание: применение этих команд может переопределить текущие настройки SNMP.

Для маршрутизатора Cisco настройка обычно включает шаги, описанные ниже.

Пример SNMPv2

1. Включите SNMP и задайте строку community.

Войдите в консоль вашего маршрутизатора Cisco и перейдите в режим настройки:

```
configure terminal
```

Затем включите SNMP, указав строку community только для чтения. Например:

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

Замените `<your_community_string>` на вашу защищенную строку community. Примечание: параметр RO (Read-Only, только чтение) позволяет SNMP получать данные с устройства, но запрещает любые изменения конфигурации.

В целях безопасности рекомендуется ограничить доступ SNMP только необходимыми устройствами. Дополнительные сведения о настройке списков контроля доступа (ACL) см. в официальной документации Cisco.

2. Сохраните конфигурацию.

Сохраните изменения, чтобы настройки SNMP сохранялись после перезагрузки:

```
write memory
```

Пример SNMPv3

SNMPv3 обеспечивает повышенную безопасность за счет аутентификации и шифрования. Его настройка более безопасна, чем SNMPv2, и должна быть проверена по документации, специфичной для вашего устройства.

1. Создайте группу SNMP.

Настройте группу SNMPv3 с включенной конфиденциальностью (шифрованием):

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

2. Создайте пользователя SNMP.

Добавьте пользователя SNMPv3 с аутентификацией и конфиденциальностью. Замените заполнители нужными вам значениями:

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

3. Сохраните конфигурацию:

```
write memory
```

Для получения дополнительных сведений или инструкций, зависящих от модели, вы можете обратиться к внешним руководствам по настройке SNMP Cisco. Однако это руководство содержит базовые шаги для включения мониторинга SNMP.

Настройка веб-интерфейса Zabbix

Создание узла сети в веб-интерфейсе Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Добавьте новый узел сети.

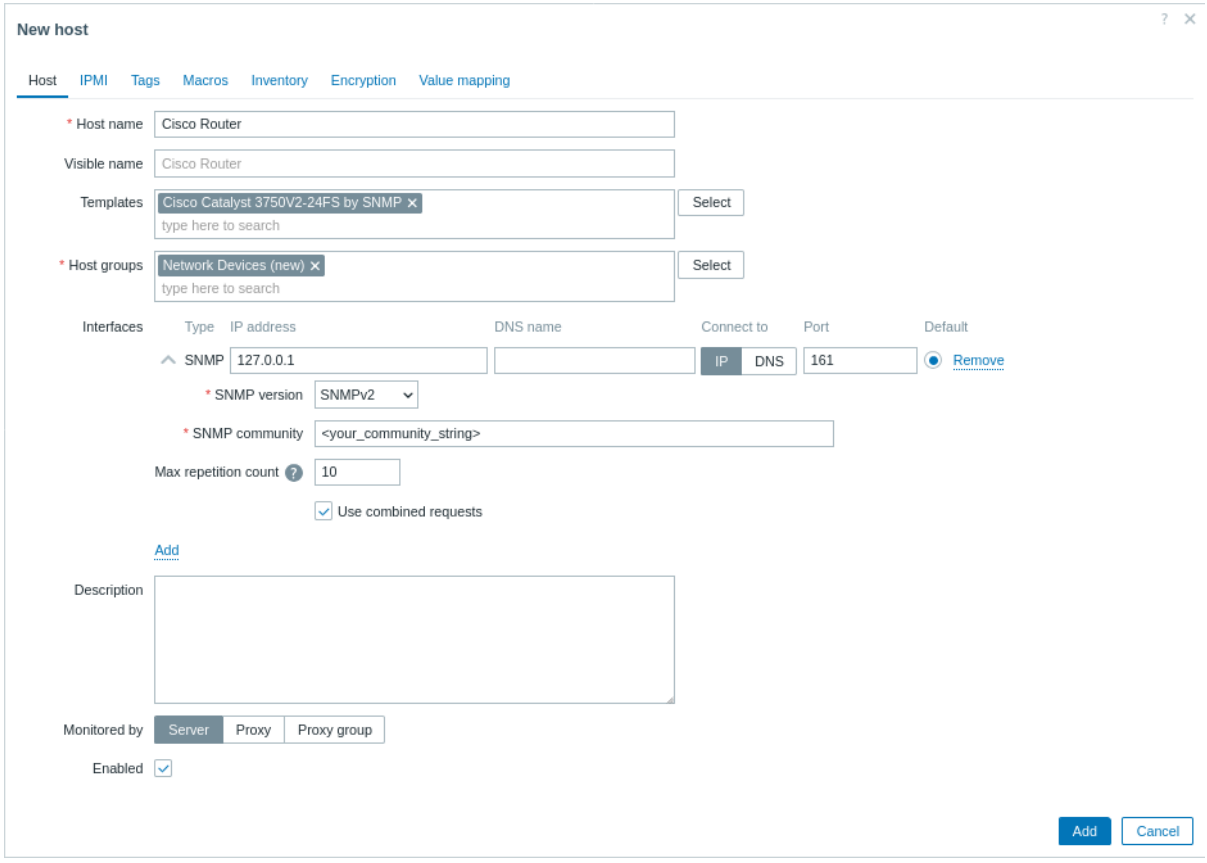
Перейдите в *Сбор данных* > *Узлы сети* и нажмите *Создать узел сети*.

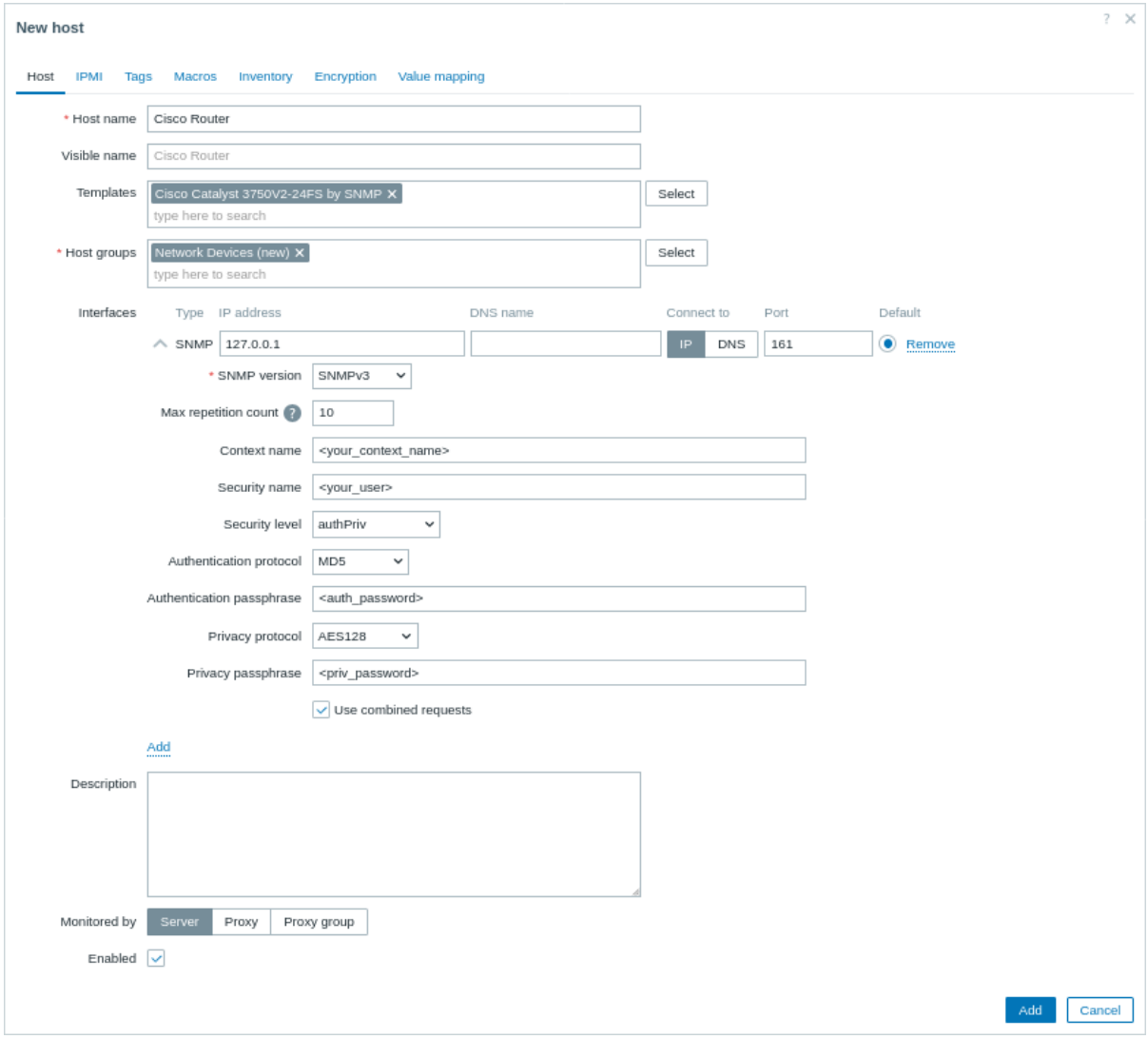
- *Имя узла сети*: введите имя для вашего устройства (например, "Cisco Router").
- Группы узлов сети: выберите существующую группу или создайте новую, например "Network Devices".
- Интерфейсы:
  - Нажмите *Добавить* в разделе Interfaces.
  - Выберите *SNMP* в качестве типа интерфейса.
  - Введите IP-адрес или DNS-имя вашего маршрутизатора Cisco.
  - Укажите порт SNMP по умолчанию (обычно 161).
  - Используйте раскрывающееся меню, чтобы выбрать подходящую версию SNMP (например, SNMPv2).
  - Для SNMPv1/v2 введите строку community в поле *SNMP community*. Для SNMPv3 будут запрошены дополнительные учетные данные (*Context name*, *Security name* и *Security level* и т. д.).

3. Свяжите шаблоны

В поле *Templates* выберите шаблон SNMP, который лучше всего соответствует вашему устройству. Zabbix предоставляет набор готовых шаблонов SNMP для многих семейств устройств. Например, если вы отслеживаете устройство Cisco, выберите шаблон, соответствующий ОС или модели вашего устройства (например, Cisco IOS SNMP или Cisco Catalyst 3750<device model> SNMP).

4. Нажмите *Add*, чтобы сохранить узел сети.



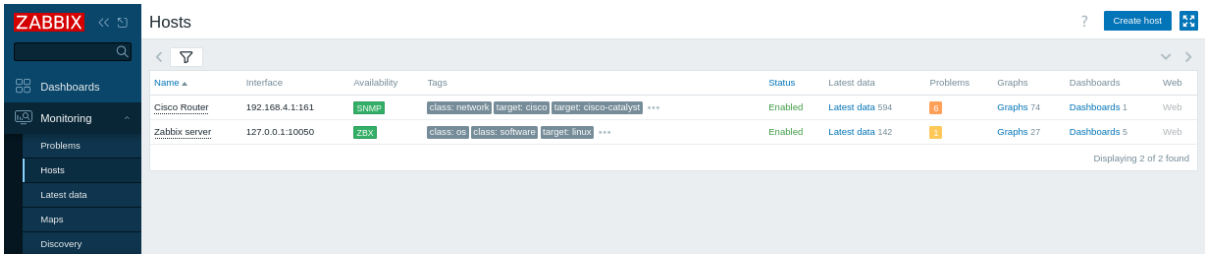


Просмотр собранных метрик

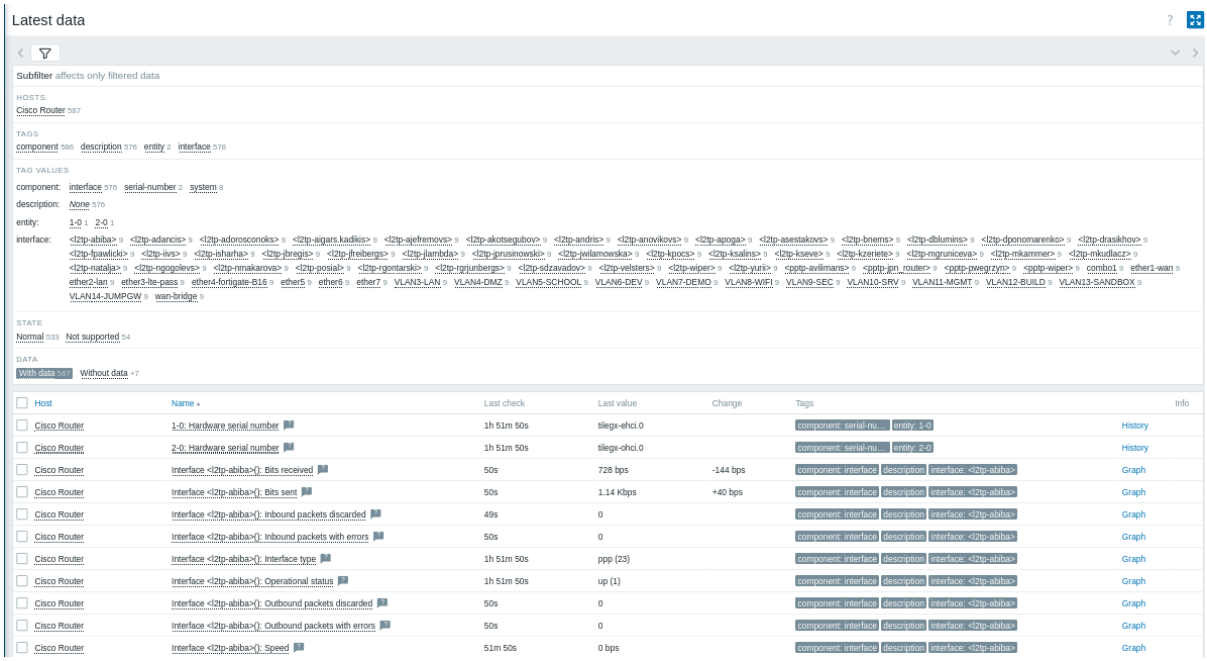
Поздравляем! Zabbix теперь настроен для мониторинга вашего сетевого устройства.

Последние данные:

- Перейдите в Monitoring > Latest data в веб-интерфейсе Zabbix.



- Выберите ваш узел сети ”Cisco Router” (или обнаруженные узлы сети), чтобы просмотреть такие метрики, как время безотказной работы оборудования и сети, потери ICMP, ping и время отклика и т. д.



- Графики и экраны:

Чтобы визуализировать данные о производительности, нажмите *Graphs* рядом с элементами данных SNMP, чтобы увидеть подробные метрики.

Следующим шагом вы можете:

- Добавить пользовательские элементы данных SNMP, чтобы отслеживать дополнительные метрики.
- Настроить оповещения о проблемах, чтобы получать уведомления о возможных неполадках.

Создание элементов данных SNMP

После настройки узла сети вы можете создать элементы данных для мониторинга конкретных метрик. Примечание: этот шаг необязателен, если вы используете шаблон, так как шаблоны уже содержат наборы элементов данных по умолчанию.

1. Определите SNMP OID:

Используйте команду `snmpwalk`, чтобы вывести список доступных OID на вашем устройстве. Например:

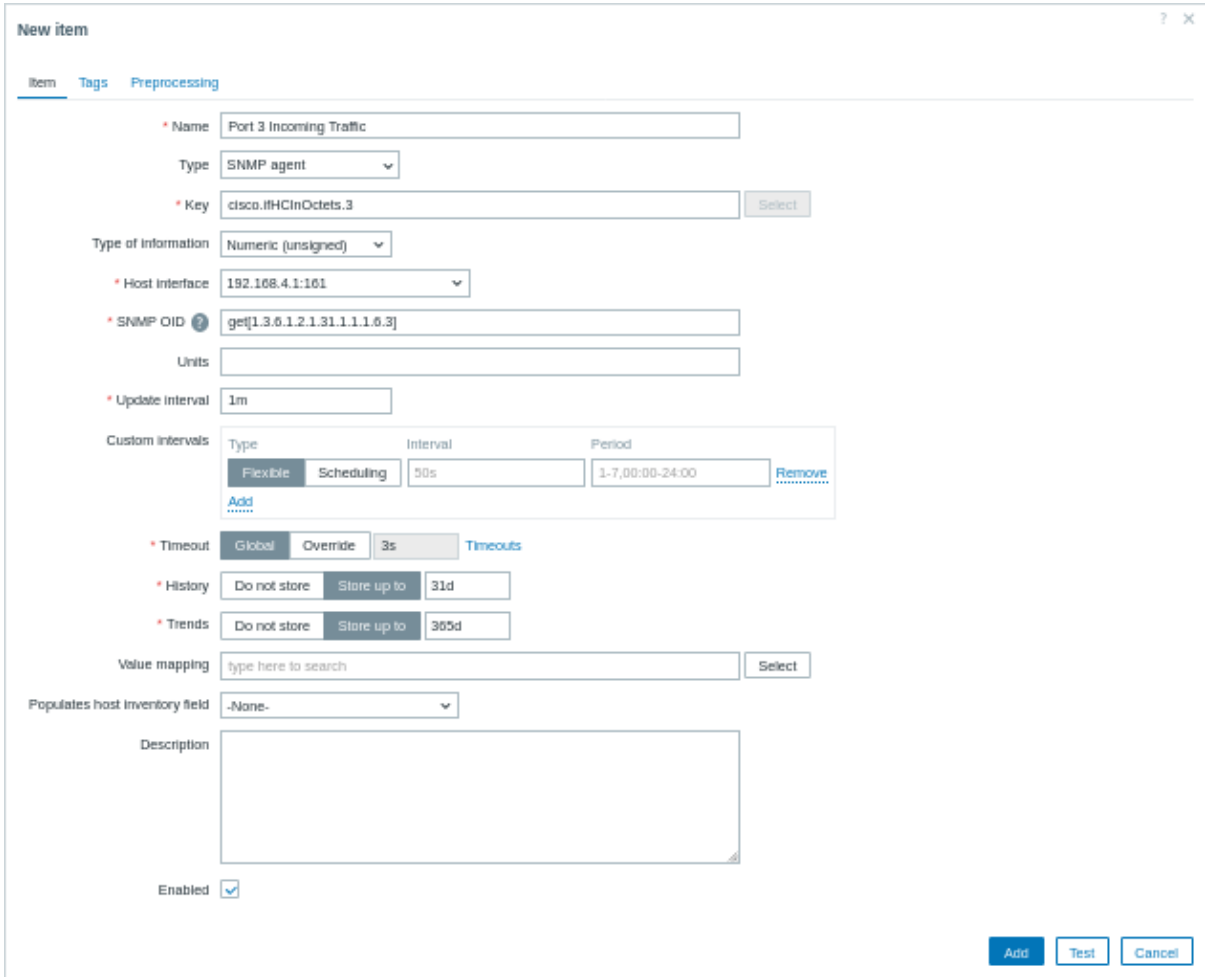
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Найдите OID для метрики, которую вы хотите отслеживать (например, IF-MIB::ifHCInOctets.3 для входящего трафика на порту 3). Чтобы получить числовой OID, можно использовать:

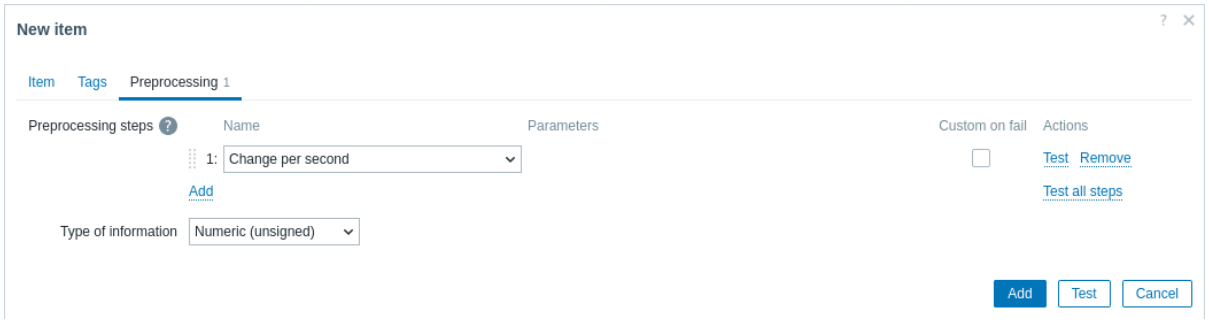
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Создайте элемент данных SNMP:

- Перейдите в *Сбор данных* > *Узлы сети* и откройте вкладку *Элементы данных* для вашего SNMP-узла сети, затем нажмите *Создать элемент данных*.
- *Имя*: укажите понятное имя (например, ”Входящий трафик на порту 3”).
- *Тип*: выберите *SNMP agent*.
- *Ключ*: укажите осмысленный ключ (например, `cisco.ifHCInOctets.3`).
- *Интерфейс узла сети*: убедитесь, что выбран SNMP-интерфейс.
- *SNMP OID*: введите OID, используя один из поддерживаемых форматов, например:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` для одного значения;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` для асинхронного получения дерева значений.



- *Предобработка* (при необходимости): если элемент данных возвращает накопительный счетчик (например, трафик интерфейса), перейдите на вкладку *Предобработка* и добавьте шаг предобработки, например "Изменение за секунду", чтобы вычислить скорость.



Чтобы получить несколько значений за одну SNMP-транзакцию, можно указать несколько OID, используя синтаксис `walk[OID1,OID2,...]`.

Перевод OID между числовым представлением и именами MIB

При работе с SNMP может потребоваться преобразование между числовыми OID и соответствующими именами MIB. Такой перевод помогает проще определять и устранять неполадки в метриках.

- Чтобы перевести имя MIB в числовой OID, используйте команду `snmptranslate` с параметром `-On`. Например, чтобы перевести имя MIB `IF-MIB::ifHCInOctets.3` в его числовой OID, выполните:

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

Эта команда может вывести:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Чтобы перевести числовой OID в имя MIB, используйте команду `snmptranslate` с параметром `-IR` (или `-m ALL`) для обратного преобразования. Например, чтобы перевести числовой OID `.1.3.6.1.2.1.31.1.1.1.6.3` обратно в его имя MIB, выполните:

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

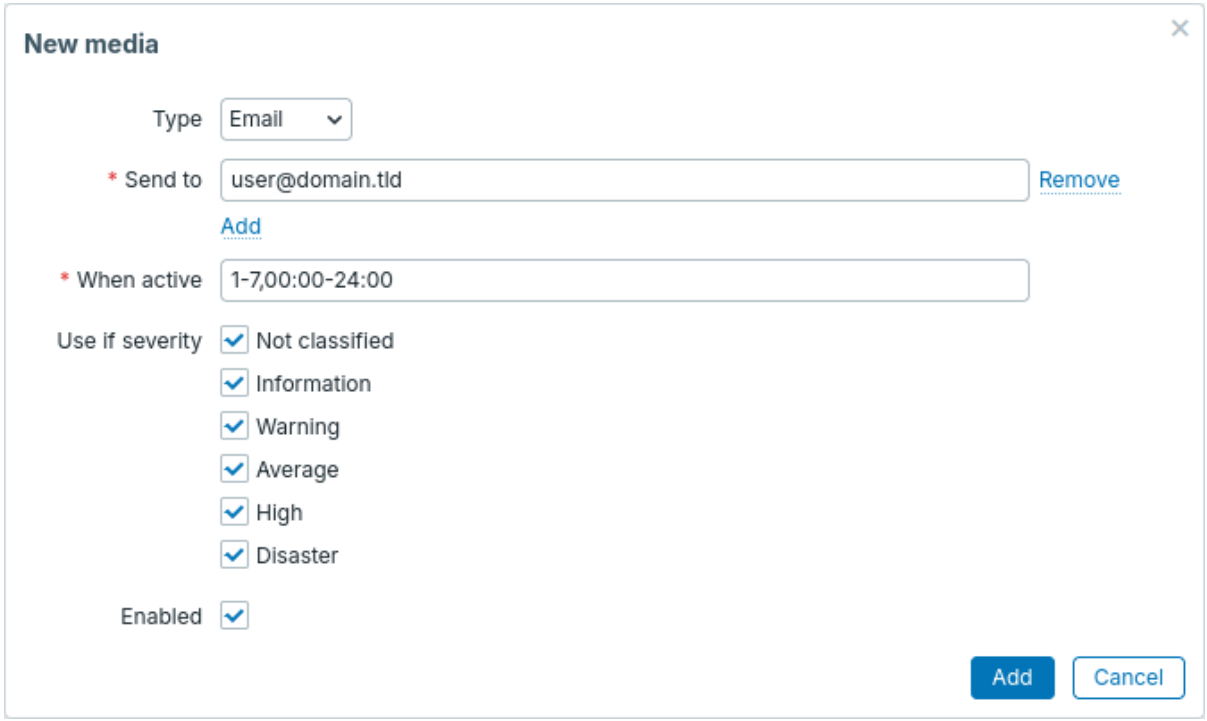
Эта команда может вывести:

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

Настройка оповещений о проблемах

Это руководство содержит основные шаги настройки отправки оповещений по электронной почте.

1. Перейдите в *User settings* > *Profile*, откройте вкладку *Media* и добавьте свой адрес электронной почты.



2. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию

Чтобы убедиться, что Zabbix корректно обнаруживает проблемы с производительностью сети, смоделируйте реальную проблему, увеличив порог времени отклика ICMP ping.

1. Откройте конфигурацию узла сети "Cisco Router" в Zabbix.

2. Перейдите на вкладку *Macros* и выберите *Inherited and host macros*.

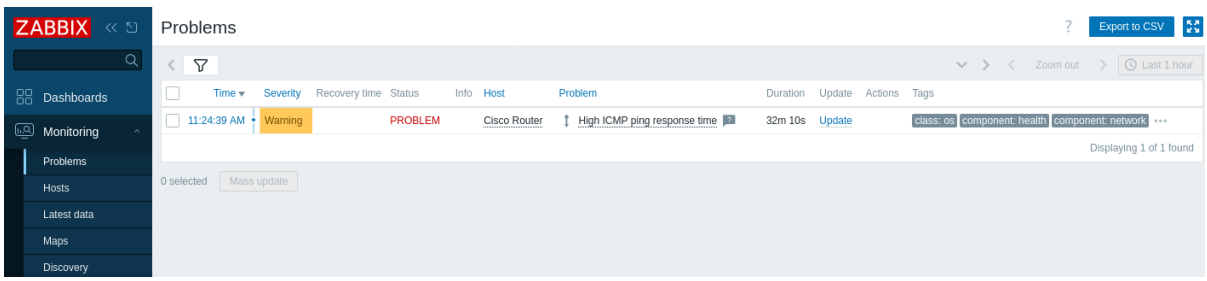
3. Найдите макрос `{$ICMP_RESPONSE_TIME_WARN}` (или аналогичный макрос порога времени отклика).

4. Установите очень низкое значение (например, 0.001), чтобы сработало оповещение, когда время отклика ping превысит это значение.

5. Нажмите *Update*, чтобы применить изменения.

6. Подождите несколько минут, пока Zabbix обнаружит смоделированную проблему.

7. Перейдите в *Monitoring* > *Problems*, чтобы убедиться, что появилось оповещение (например, "High ICMP ping response time").



Если оповещения настроены, вы также должны получить оповещение о проблеме.

8. Верните значение макроса к исходному и нажмите *Update*, чтобы сохранить изменения.

9. Убедитесь, что проблема устранена и исчезла из раздела *Problems*.

Устранение неполадок при мониторинге SNMP

Если вы заметили, что значок SNMP в веб-интерфейсе Zabbix отображается КРАСНЫМ или данные не собираются, попробуйте выполнить следующие шаги:

1. Проверьте подключение по SNMP.

Для SNMPv2 выполните следующую команду с вашего сервера Zabbix:

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

Эта команда проверяет, отвечает ли устройство на SNMP-запросы.

Для SNMPv3 укажите соответствующие учетные данные SNMPv3:

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

Это подтверждает, что учетные данные SNMPv3 верны и устройство отвечает безопасным образом.

2. Убедитесь, что файлы MIB установлены и включены, как описано в предварительных требованиях. Чтобы убедиться в этом, следующая команда не должна выдавать ошибку при запросе сетевого устройства:

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

Она должна возвращать преобразованные OID без ошибок.

3. Убедитесь, что версия SNMP и учетные данные, настроенные в Zabbix, совпадают с теми, что заданы на вашем устройстве. Например, проверьте настройки SNMP в конфигурации узла сети Zabbix и сравните их с конфигурацией вашего устройства. На устройстве Cisco можно проверить настройки SNMP, выполнив:

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

Это гарантирует, что строка community (для SNMPv2) или сведения о пользователе SNMPv3 указаны верно.

4. Убедитесь, что SNMP корректно включен на вашем сетевом устройстве. На маршрутизаторе Cisco войдите в консоль и выполните:

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

Эта команда отображает активную конфигурацию SNMP и помогает подтвердить, что SNMP настроен правильно.

5. Убедитесь, что брандмауэры или сетевые проблемы не блокируют SNMP-трафик (обычно порт 161) между сервером Zabbix и устройством. Проверить подключение можно с помощью:

nc -zv <device_IP> 161

`nc -zv` проверяет, открыт ли порт 161 и прослушивается ли он на устройстве.

Кроме того, если вы используете UFW в Ubuntu, проверьте состояние брандмауэра:

```
sudo ufw status
```

Или для iptables:

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

6. Просмотрите файлы журнала сервера Zabbix на наличие ошибок, связанных со SNMP, чтобы точнее определить проблему:

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

`tail -f` позволяет отслеживать обновления журнала в реальном времени.

См. также:

- Создание элемента данных - узнайте, как добавить дополнительные метрики.
- SNMP-агент - дополнительная информация о мониторинге SNMP с помощью Zabbix.
- Стандартизированные шаблоны для сетевых устройств - информация о доступных шаблонах SNMP.
- Обнаружение SNMP OID - дополнительная информация об обнаружении SNMP на коммутаторе.
- Настройка правила сетевого обнаружения - дополнительная информация о том, как настроить правило сетевого обнаружения, используемое Zabbix для обнаружения узлов сети и служб.

Мониторинг журнала событий Windows с использованием активных проверок

Введение

Это руководство объясняет, как отслеживать журналы событий Windows с помощью Zabbix, используя активные проверки. С помощью ключей элементов данных Zabbix, специфичных для Windows, вы можете собирать и анализировать критические события (например, неудачные попытки входа в систему, системные ошибки и т. д.) в реальном времени.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и сетевых администраторов, которые хотят отслеживать журналы событий Windows. Для получения сведений о дополнительных параметрах настройки см. документацию по ключам элементов данных, специфичным для Windows.

Предварительные требования

Перед тем как продолжить работу с этим руководством, вам необходимо скачать и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС. Вам также необходимо, чтобы агент Zabbix был скачан и установлен на компьютере с Windows, который вы хотите отслеживать.

Настройка агента Zabbix для мониторинга журнала событий Windows

1. Откройте файл конфигурации агента Zabbix на узле сети Windows и убедитесь, что параметр `ServerActive` задан как IP-адрес вашего сервера Zabbix, а параметр `Hostname` соответствует имени узла сети, которое будет указано в веб-интерфейсе Zabbix. Это позволяет агенту запрашивать активные проверки для своего узла сети с указанного сервера Zabbix. Например:

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

2. Перезапустите службу агента Zabbix, чтобы применить изменения:

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

3. Проверьте, что узел сети Windows работает:

- Убедитесь, что служба агента Zabbix запущена на узле сети Windows.
- Проверьте, что узел сети Windows может подключиться к серверу Zabbix через порт 10051. Чтобы проверить подключение с узла сети Windows, откройте PowerShell и выполните следующую команду:

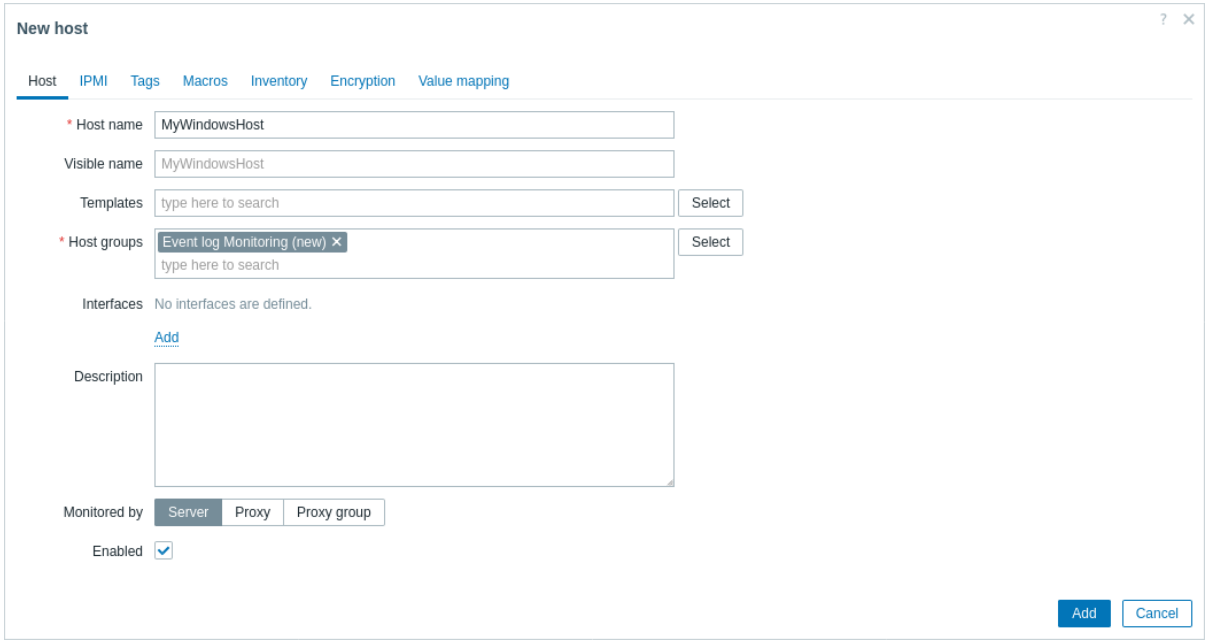
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Настройка веб-интерфейса Zabbix

1. Перейдите в *Сбор данных* > *Узлы сети* и создайте узел сети:

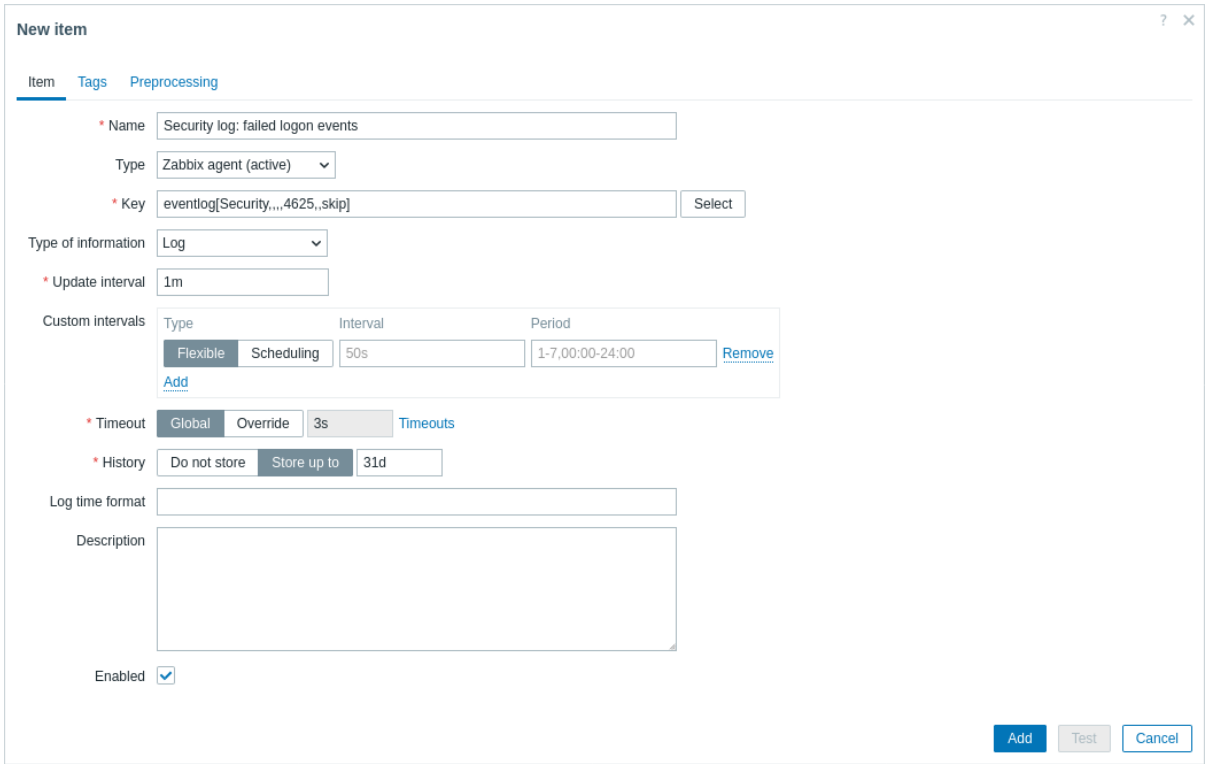
- В поле *Имя узла сети* введите имя узла сети (например, ”MyWindowsHost”).
- В поле *Группы узлов сети* введите или выберите группу узлов сети (например, ”Event log Monitoring”).
- Нажмите *Добавить*, чтобы сохранить настроенный узел сети.

> **Note:**
> В поле *Шаблоны* вы можете добавить шаблон ”Windows by Zabbix agent active”, чтобы помочь в поиске неисправностей, наблюдая, обновляются ли другие активные элементы данных на том же узле сети.



2. Создайте новый элемент данных со следующими параметрами:

- В поле *Имя* введите описательное имя элемента данных (например, ”Security log: failed logon events”).
- В раскрывающемся списке *Тип* выберите ”Zabbix agent (active)” (требуется для мониторинга журнала событий).
- В поле *Ключ* используйте ключ элемента данных eventlog. Например, чтобы отслеживать неудачные попытки входа в систему (Event ID: 4625) в журнале Security и игнорировать записи, более старые, чем последняя проверка элемента данных (с использованием параметра `skip`), введите следующий ключ элемента данных: `eventlog[Security,,,,4625,,skip]`
- В раскрывающемся списке *Тип информации* выберите ”Log”.



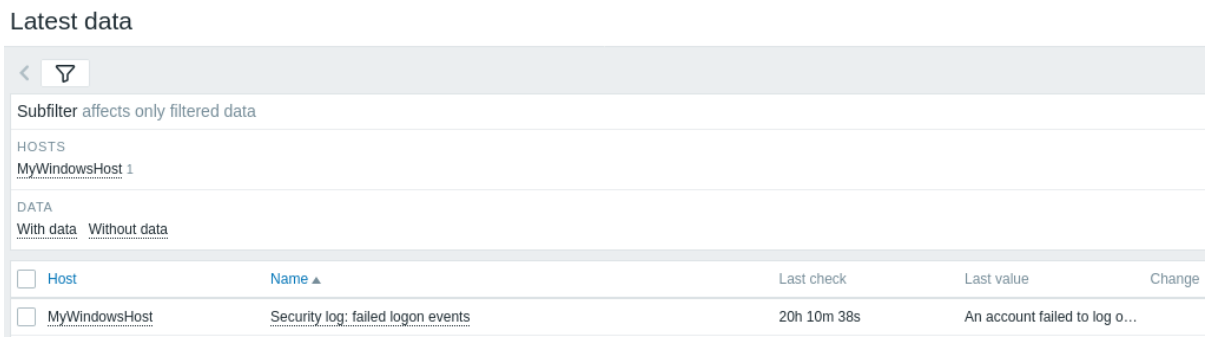
3. Нажмите *Добавить*, чтобы сохранить элемент данных.

Тестирование и просмотр собранных метрик

Поздравляем! Теперь Zabbix настроен на сбор журналов событий Windows. Чтобы убедиться, что журналы событий собираются, вы можете протестировать элемент данных ”Security log: failed logon events”, выйдя из своей учетной записи Windows и попытавшись войти с неверными учетными данными.

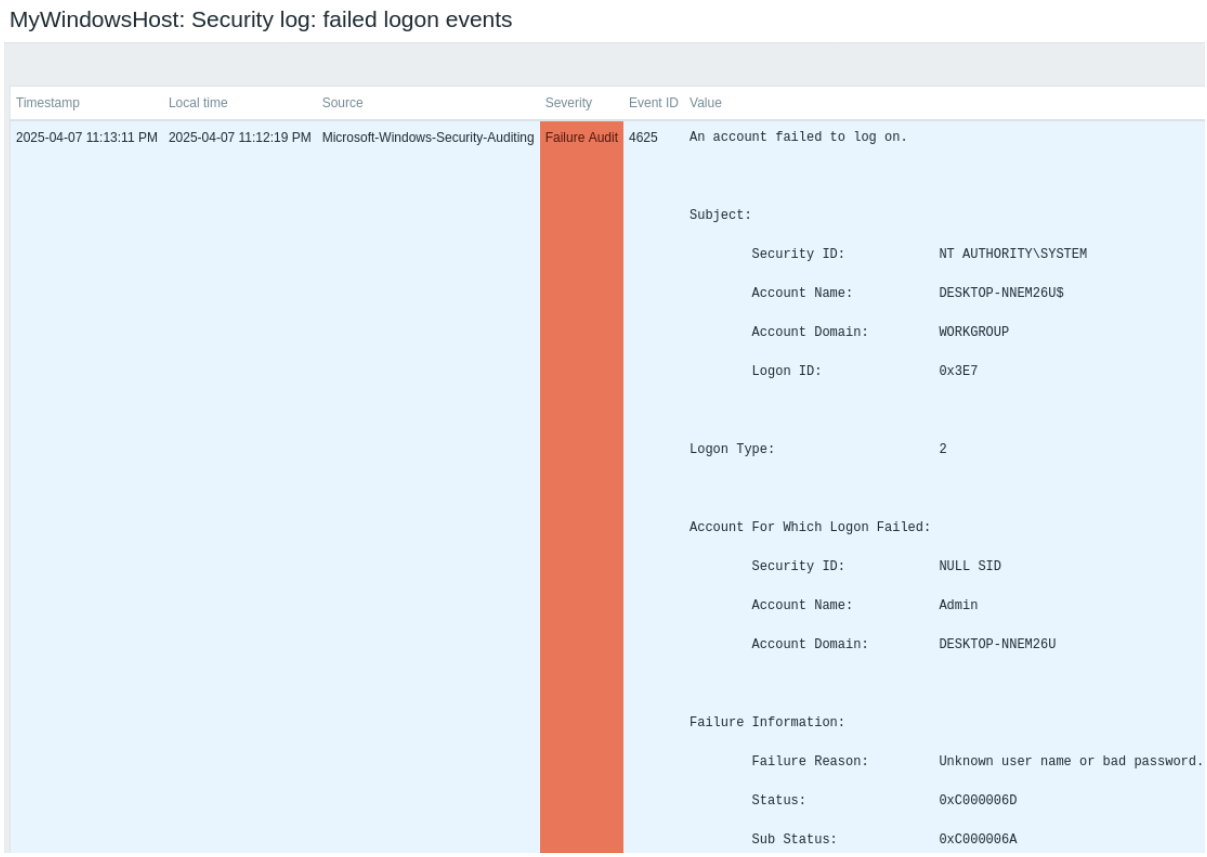
Затем просмотрите собранные журналы в веб-интерфейсе Zabbix:

1. Перейдите в *Monitoring* > *Latest data* в веб-интерфейсе Zabbix.



2. Отфильтруйте по вашему узлу сети ”MyWindowsHost” в поле *Name*.

3. Нажмите *History*, чтобы просмотреть записанные значения журнала.



4. Если значения журнала отсутствуют, перейдите к разделу Troubleshooting данного руководства.

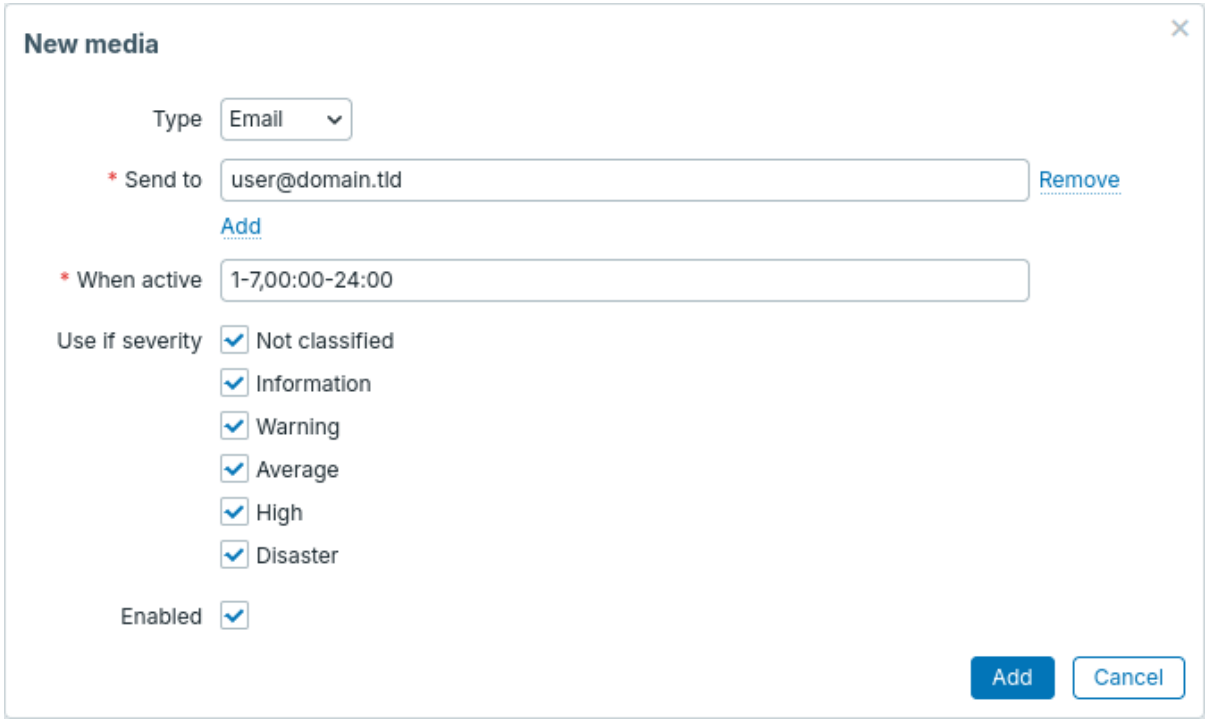
Настройка оповещений о проблемах

В этом руководстве приведены основные шаги настройки для отправки оповещений по электронной почте.

1. Перейдите в Data collection > Узлы сети, чтобы определить триггер, который срабатывает, когда элемент данных журнала событий записывает интересующий вас шаблон. Например, чтобы отлавливать неудачные попытки входа в журнале Security, используйте функцию find():

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

2. Перейдите в *User settings > Profile*, откройте вкладку *Media* и добавьте свой email.



3. Следуйте руководству по получению уведомления о проблеме.

В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Устранение неполадок

Если у вас возникают проблемы со сбором или просмотром журналов событий Windows, воспользуйтесь приведенными ниже советами, чтобы выявить и устранить распространенные проблемы:

1. На сервере Zabbix (Linux) выполните следующую команду, чтобы вывести правила iptables:

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

и убедитесь, что для TCP-порта 10051 есть правило ACCEPT.

2. Убедитесь, что ключ `eventlog[...]` использует точное имя журнала (с учетом регистра), идентификатор события, режим (например, skip) и другие параметры в точности так, как показано в ключах элементов данных, специфичных для Windows.

См. также:

- Создание элемента данных - узнайте, как добавлять дополнительные метрики.
- Агент Zabbix в Microsoft Windows - подробные инструкции по установке.
- Мониторинг Windows с помощью агента Zabbix - подробное руководство по настройке базового мониторинга для машин Windows с использованием агента Zabbix.
- Ключи элементов данных, специфичные для Windows - подробная информация о ключах элементов данных, специфичных для Windows, поддерживаемых агентами Zabbix, включая ключи для мониторинга журнала событий.
- Мониторинг файлов журналов - инструкции по настройке Zabbix для централизованного мониторинга и анализа файлов журналов, применимые к журналам событий Windows.

Установка: дополнительные руководства

Если вы ищете самый простой способ установки Zabbix, см. раздел Установка и первые шаги.

В приведенных ниже руководствах описаны **дополнительные** или специфичные для системы способы установки Zabbix или его компонентов:

- Установка из контейнеров
- Установка из исходных кодов
- Примечания по установке из пакетов
- Установка агента Zabbix из MSI-пакета для Windows
- Сборка агента Zabbix 2 в Windows
- Сборка агента Zabbix в Windows

- Установка агента в macOS из PKG-пакета
- Агент Zabbix в IBM AIX
- Сборка агента Zabbix в macOS
- Аппаратное устройство Zabbix
- Руководство по развертыванию сервера Zabbix в AWS
- Руководство по развертыванию прокси Zabbix в AWS
- Руководство по развертыванию сервера Zabbix в Azure
- Руководство по развертыванию прокси Zabbix в Azure
- Руководство по развертыванию в Google Cloud

Установка из контейнеров

Обзор

На этой странице описывается, как развернуть Zabbix с помощью Docker или Docker Compose.

Docker Compose — самый быстрый способ развернуть Zabbix. Он читает файл конфигурации и автоматически запускает все контейнеры, из которых состоит полная установка Zabbix, в правильном порядке.

Docker (ручное развертывание) позволяет добиться того же результата, развертывая каждый компонент шаг за шагом.

Предварительные требования

Перед началом убедитесь, что в вашей системе установлен Docker версии 1.12.0 или более поздней. Если Docker не установлен, воспользуйтесь руководством по установке Docker. Для развертывания с помощью Docker Compose также требуется Docker Compose версии 2.24.0 или более поздней.

Для некоторых компонентов Zabbix необходимо открыть определенные порты на узле сети, на котором запущен Docker (например, 10051/TCP для сервера Zabbix и 162/UDP для SNMP-трапов). Полный список портов, используемых компонентами Zabbix, см. в разделе Требования. Для сервера Zabbix и агента порт по умолчанию можно изменить, задав переменную окружения `ZBX_LISTENPORT` в соответствующем контейнере.

Доступные образы Docker

Zabbix предоставляет образ Docker для каждого компонента Zabbix. Все образы опубликованы на Docker Hub. Каждый образ используется для создания контейнера, в котором запускается соответствующий компонент.

Компонент Zabbix	Образ Docker
Агент	zabbix/zabbix-agent
Агент 2	zabbix/zabbix-agent2
Сервер (MySQL)	zabbix/zabbix-server-mysql
Сервер (PostgreSQL)	zabbix/zabbix-server-pgsql
Веб-интерфейс (Apache + MySQL)	zabbix/zabbix-web-apache-mysql
Веб-интерфейс (Apache + PostgreSQL)	zabbix/zabbix-web-apache-pgsql
Веб-интерфейс (Nginx + MySQL)	zabbix/zabbix-web-nginx-mysql
Веб-интерфейс (Nginx + PostgreSQL)	zabbix/zabbix-web-nginx-pgsql
Прокси (SQLite3)	zabbix/zabbix-proxy-sqlite3
Прокси (MySQL)	zabbix/zabbix-proxy-mysql
Java gateway	zabbix/zabbix-java-gateway
Веб-сервис	zabbix/zabbix-web-service
Ловушки SNMP	zabbix/zabbix-snmptraps
Сервер MCP	zabbix/zabbix-mcp-server

Note:
Для использования ловушек SNMP контейнер с ловушками SNMP должен совместно использовать том с контейнером сервера или прокси Zabbix (см. раздел как использовать этот образ на Docker Hub и приведенный ниже пример).

Все образы компонентов Zabbix на Docker Hub основаны на последних основных версиях поддерживаемых операционных систем. Эти образы автоматически пересобираются при обновлении базовых образов ОС.

Теги образов

Каждый образ поддерживает теги для выбора базовой операционной системы и версии Zabbix:

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

Поддерживаемые значения <os>:

- alpine - Alpine Linux
- ubuntu - Ubuntu
- centos - CentOS Stream
- ol - Oracle Linux
- ltsc2022 - Windows 11 LTSC 2022 (только Zabbix агент)

Поддерживаемые значения <version>:

- latest - Последняя стабильная версия Zabbix на Alpine Linux
- <os>-latest - Последняя стабильная версия Zabbix на выбранной ОС
- <os>-trunk - Последняя разрабатываемая (nightly) сборка на выбранной ОС
- <os>-X.X-latest - Последний минорный релиз Zabbix для указанной мажорной версии Zabbix на выбранной ОС
- <os>-X.X.* - Конкретный минорный релиз Zabbix на выбранной ОС

Примеры:

```
####              Zabbix      (SQLite3)    Alpine Linux:
docker pull zabbix/zabbix-proxy-sqlite3:latest

####              Zabbix      (SQLite3)    Ubuntu:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

####                 (nightly)      Zabbix      (MySQL)    Ubuntu:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk

####                 8.0 Zabbix      (MySQL)    Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

####      8.0.1 Zabbix      (MySQL)    Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose - это самый быстрый способ развернуть Zabbix. Он считывает файл конфигурации (файл Compose) и выполняет всю настройку: загружает образы Docker, создает внутреннюю сеть между контейнерами, настраивает хранилище, инициализирует базу данных и запускает все компоненты в правильном порядке.

Репозиторий официальных Dockerfile Zabbix содержит готовые к использованию файлы Docker Compose и систему конфигурации на основе .env, поддерживающую различные операционные системы, серверные части баз данных и настройку компонентов Zabbix.

1. Клонируйте репозиторий, перейдите в него и переключитесь на версию 8.0:

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

2. (Необязательно) Настройте развертывание с помощью переменных окружения. Этот шаг можно пропустить для настройки по умолчанию.

3. Разверните настройку по умолчанию: сервер Zabbix, веб-интерфейс (работающий на NGINX) и серверную часть базы данных (MySQL или PostgreSQL), каждый компонент запускается в собственном контейнере на Alpine Linux.

```
####   MySQL               :
docker compose -f ./compose.yaml up -d

####   PostgreSQL               :
docker compose -f ./compose_pgsql.yaml up -d
```

После запуска контейнеров (обычно в течение 1-3 минут) сервер Zabbix запускается, а веб-интерфейс становится доступен для входа.

Note:
Используйте `docker compose ps`, чтобы проверить состояние контейнеров. Все контейнеры (за исключением контейнера `zabbix-docker-server-db-init-1`) должны иметь статус `Up`. Если у какого-либо контейнера отображается статус `Exited`, проверьте его журналы на наличие ошибок с помощью команды `docker logs -f <container-name>`.

Переменные окружения

Поведение Docker Compose и конфигурацию компонентов Zabbix можно настраивать с помощью переменных окружения.

Переменные уровня Compose (определенные в файле `.env`) управляют тем, какие образы Docker, порты и диапазоны IP-адресов сети используются. Эти переменные можно указать в командной строке перед командой `docker compose` или изменить в файле `.env`.

Например, следующая команда развертывает полную многоконтейнерную конфигурацию Zabbix с использованием образа на базе Ubuntu (вместо Alpine Linux по умолчанию) и предоставляет доступ к веб-интерфейсу через NGINX на пользовательских портах HTTP (8282) и HTTPS (8443):

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

Переменные уровня компонентов (определенные в файлах `env_vars/.env_<component>`) управляют конфигурацией компонентов Zabbix. Измените соответствующий файл `.env_<component>` перед выполнением команды `docker compose`.

Например, можно настроить количество пассивных опросчиков сервера Zabbix и параметры кэша, а также часовой пояс веб-интерфейса, изменив следующие переменные:

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

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

Note:
Переменные окружения уровня компонентов соответствуют параметрам конфигурации компонентов Zabbix, но используют другой стиль именования (например, `ZBX_STARTPOLLERS` соответствует `StartPollers`). Некоторые переменные относятся только к Docker, а некоторые параметры конфигурации изменить нельзя (например, `PIDFile` и `LogType`). При использовании переменных окружения см. раздел *Environment variables* в описании образа Docker каждого компонента на Docker Hub.

Тома

Docker Compose хранит постоянные данные в каталоге `zbx_env/`, создаваемом рядом с файлом Compose. Этот каталог сохраняет данные компонентов при перезапуске и обновлении контейнеров.

Содержимое `zbx_env/` заранее определено для образа каждого компонента. Например:

- `zbx_env/etc/` — используется контейнером веб-интерфейса Zabbix
- `zbx_env/usr/` и `zbx_env/var/` — используются контейнером сервера Zabbix

Дополнительную информацию о каждом томе см. в разделе *Allowed volumes* в описании Docker image каждого компонента на Docker Hub.

Makefile

Репозиторий Dockerfiles Zabbix также предоставляет Makefile для упрощения выполнения распространённых задач Docker Compose. Вместо полной команды `docker compose` можно использовать более короткие команды `make` (выполните `make help`, чтобы просмотреть доступные параметры):

```
#### (    Zabbix,   -     , MySQL —      Alpine Linux)
make up

#### (    Zabbix,   -                    NGINX, PostgreSQL —      Ubuntu)
make up \
```

```
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

Attention:
При остановке или удалении контейнеров всегда указывайте тот же тип базы данных, который использовался при развёртывании (например, `make down DB=pgsql`).

По умолчанию команда `make up` запускает только минимальный набор служб (сервер Zabbix, веб-интерфейс и базу данных). Это позволяет сохранить конфигурацию лёгкой и не запускать компоненты, которые могут не потребоваться. Чтобы включить дополнительные компоненты, можно использовать следующие профили Compose:

```
#### + Zabbix, Java gateway, - SNMP:
make up COMPOSE_PROFILES=full

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

Docker (ручное развертывание)

Используйте ручное развертывание, если хотите развернуть Zabbix поэтапно, запускать отдельные компоненты, интегрировать их с существующей средой или использовать альтернативную среду выполнения контейнеров, например Podman.

Например, чтобы развернуть прокси Zabbix, выполните следующую команду:

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

Эта команда:

- Загружает Docker-образ `zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest`.
- Создает и запускает контейнер `zabbix-proxy-sqlite3` на основе Docker-образа (с флагами `--init` и `-d`).
- Настраивает параметр Zabbix proxy `Server` через переменную среды `ZBX_SERVER_HOST`. Этот параметр задает IP-адрес сервера Zabbix, с которого прокси будет получать данные конфигурации и на который будет отправлять собранные данные. Обратите внимание, что другие переменные, необходимые для работы активного прокси, используют значения по умолчанию и могут быть опущены.

Note:
Переменные среды соответствуют параметрам конфигурации компонентов Zabbix, но используют другой стиль именования (например, `ZBX_SERVER_HOST` соответствует `Server`). Некоторые переменные являются специфичными для Docker, а некоторые параметры конфигурации изменить нельзя (например, `PIDFile` и `LogType`). При использовании переменных среды, пожалуйста, обращайтесь к разделу *Environment variables* в описании каждого Docker-образа компонента на Docker Hub.

- Подключает управляемое Docker хранилище (например, `/var/lib/docker/volumes/zabbix-proxy-data/`) к каталогу `/var/lib/zabbix` контейнера, чтобы данные прокси Zabbix сохранялись постоянно, даже если контейнер будет удален.

Note:
Дополнительные сведения о каждом томе см. в разделе *Allowed volumes* в описании каждого Docker-образа компонента на Docker Hub.

После развертывания контейнера прокси Zabbix вы можете перейти к добавлению прокси в ваш веб-интерфейс Zabbix и настройке узлов сети для мониторинга через этот прокси.

Примеры ниже охватывают еще три сценария развертывания:

- сервер Zabbix (MySQL) с Java gateway
- сервер Zabbix (PostgreSQL) с SNMP traps
- сервер Zabbix (MySQL) с Java gateway на RHEL 8-10

Дополнительные примеры см. в описании каждого Docker-образа компонента на Docker Hub.

Развертывание сервера Zabbix (MySQL) с Java gateway

В этом примере показано, как развернуть сервер Zabbix с серверной частью MySQL, Java gateway для мониторинга JMX и веб-интерфейсом на базе NGINX.

1. Создайте выделенную сеть Docker, чтобы все контейнеры компонентов Zabbix могли обращаться друг к другу по имени контейнера:

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

2. Запустите пустой контейнер сервера MySQL (замените `zabbix_pwd` и `root_pwd` на надежные пароли и используйте эти значения на следующих шагах):

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

3. Просматривайте журналы MySQL и дождитесь готовности MySQL принимать подключения (для выхода из просмотра журналов нажмите `Ctrl+C`):

```
docker logs -f mysql-server

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

4. Запустите контейнер Java gateway Zabbix:

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

5. Включите `log_bin_trust_function_creators` в контейнере сервера MySQL. Это временно ослабляет ограничения безопасности для хранимых функций; на следующих шагах этот параметр будет отключен. Это позволяет пользователю базы данных Zabbix создавать хранимые функции без привилегии `SUPER`, которая требуется MySQL при включенном двоичном журнале (по умолчанию начиная с MySQL 8.0). Выполните следующую команду:

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

6. Запустите контейнер сервера Zabbix и свяжите его с контейнерами Java Gateway и сервера MySQL:

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

7. Просматривайте журналы сервера Zabbix и дождитесь завершения инициализации схемы базы данных сервером Zabbix (для выхода из просмотра журналов нажмите `Ctrl+C`):

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

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

8. Отключите log_bin_trust_function_creators в контейнере сервера MySQL. Это повторно включает ограничение безопасности, запрещающее пользователям без привилегии SUPER создавать хранимые функции. Выполните следующую команду:

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

9. Запустите контейнер веб-интерфейса Zabbix и свяжите его с контейнерами сервера Zabbix и сервера MySQL:

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

После запуска контейнеров сервер Zabbix запускается, а веб-интерфейс становится доступен для входа.

> **Note:**
> Используйте docker ps, чтобы проверить состояние контейнеров. Все контейнеры должны иметь состояние Up. Если у какого-либо контейнера отображается состояние Exited, проверьте его журналы на наличие ошибок с помощью docker logs -f <container-name>.

Развертывание сервера Zabbix (PostgreSQL) с SNMP-ловушками

В этом примере показано, как развернуть сервер Zabbix с серверной частью PostgreSQL, SNMP-ловушками и веб-интерфейсом на базе NGINX.

1. Создайте выделенную сеть Docker, чтобы все контейнеры компонентов Zabbix могли обращаться друг к другу по имени контейнера:

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

2. Запустите пустой контейнер сервера PostgreSQL (замените zabbix_pwd на надежный пароль и используйте это значение на следующих шагах):

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

3. Запустите контейнер SNMP-ловушек Zabbix:

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

4. Запустите контейнер сервера Zabbix и свяжите его с контейнерами сервера PostgreSQL и SNMP-ловушек:

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

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

5. Просматривайте журналы сервера Zabbix и дождитесь завершения инициализации схемы базы данных сервером Zabbix (для выхода из просмотра журналов нажмите Ctrl+C):

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

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

6. Запустите контейнер веб-интерфейса Zabbix и свяжите его с контейнерами сервера Zabbix и сервера PostgreSQL:

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

После запуска контейнеров сервер Zabbix запускается, а веб-интерфейс становится доступен для входа.

> **Note:**
> Используйте `docker ps`, чтобы проверить состояние контейнеров. Все контейнеры должны иметь состояние Up. Если у какого-либо контейнера отображается состояние Exited, проверьте его журналы на наличие ошибок с помощью `docker logs -f <container-name>`.

Развертывание сервера Zabbix (MySQL) с Java gateway в RHEL 8-10

В этом примере показано, как развернуть сервер Zabbix с серверной частью MySQL, Java gateway для мониторинга JMX и веб-интерфейсом на базе NGINX, работающие в Red Hat Enterprise Linux 8, 9 или 10.

В Red Hat Enterprise Linux рекомендуется использовать Podman вместо Docker в качестве среды выполнения контейнеров. Podman работает аналогично Docker, но не требует фоновой службы, работающей от имени root, поэтому лучше подходит для сред Red Hat.

1. Создайте новый pod с именем zabbix и опубликуйте порты для веб-интерфейса Zabbix и траппера сервера Zabbix:

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

2. (Необязательно) Запустите контейнер агента Zabbix в pod zabbix:

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

3. Создайте каталог `./mysql/` на узле RHEL:

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

4. Запустите пустой контейнер сервера MySQL (замените zabbix_pwd и root_pwd на надежные пароли и используйте эти значения на следующих шагах):

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

5. Просматривайте журналы MySQL и дождитесь готовности MySQL принимать подключения (для выхода из просмотра журналов нажмите Ctrl+C):

```
podman logs -f mysql-server

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

6. Запустите контейнер Java gateway Zabbix:

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

7. Включите log_bin_trust_function_creators в контейнере сервера MySQL. Это временно ослабляет ограничения безопасности для хранимых функций; на следующих шагах этот параметр будет отключен. Это позволяет пользователю базы данных Zabbix создавать хранимые функции без привилегии SUPER, которая требуется MySQL при включенном двоичном журнале (по умолчанию начиная с MySQL 8.0). Выполните следующую команду:

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

8. Запустите контейнер сервера Zabbix и свяжите его с контейнерами Java Gateway и сервера MySQL:

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

9. Просматривайте журналы сервера Zabbix и дождитесь завершения инициализации схемы базы данных сервером Zabbix (для выхода из просмотра журналов нажмите Ctrl+C):

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

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

10. Отключите log_bin_trust_function_creators в контейнере сервера MySQL. Это повторно включает ограничение безопасности, запрещающее пользователям без привилегии SUPER создавать хранимые функции. Выполните следующую команду:

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

11. Запустите контейнер веб-интерфейса Zabbix и свяжите его с контейнерами сервера Zabbix и сервера MySQL:

```
podman run --name zabbix-web-mysql -t \
  -e ZBX_SERVER_HOST="127.0.0.1" \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
```

```
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e PHP_TZ="Europe/Riga" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-80
```

Note:
Pod `zabbix` публикует порт 80/TCP (HTTP) на хост-компьютер из порта 8080/TCP контейнера `zabbix-web-mysql`.

После запуска контейнеров сервер Zabbix запускается, а веб-интерфейс становится доступен для входа.

Note:
Используйте `docker ps`, чтобы проверить состояние контейнеров. Все контейнеры должны иметь состояние `Up`. Если у какого-либо контейнера отображается состояние `Exited`, проверьте его журналы на наличие ошибок с помощью `docker logs -f <container-name>`.

Установка из исходного кода

Вы можете получить самую последнюю версию Zabbix, скомпилировав её из исходного кода. См. также Получение исходного кода Zabbix.

Здесь представлено пошаговое руководство по установке Zabbix из исходного кода.

Установка демонов Zabbix

1 Загрузите архив с исходным кодом

Перейдите на страницу загрузки Zabbix и загрузите архив с исходным кодом. После загрузки распакуйте исходные файлы, выполнив:

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

Note:
Укажите в команде правильную версию Zabbix. Она должна соответствовать имени загруженного архива.

2 Создание учетной записи пользователя

Все процессы демонов Zabbix запускаются от имени непривилегированных системных пользователей. Если демон Zabbix запущен из-под непривилегированной учетной записи пользователя, он продолжит работу от имени этого пользователя.

В конфигурации по умолчанию, если демон запускается от имени `root`, он переключается на учетную запись пользователя `zabbix`, которая должна существовать. Чтобы создать системного пользователя и группу `zabbix`, выполните команды, приведенные ниже.

Система на базе RedHat:

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

Система на базе Debian:

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

Нет необходимости создавать отдельную учетную запись пользователя для веб-интерфейса Zabbix.

Рекомендация по безопасности

Если Zabbix сервер и агент работают на одной и той же машине, рекомендуется запускать их от имени **разных учетных записей пользователей**. Запуск обоих от имени одного и того же пользователя позволяет агенту получить доступ к файлу конфигурации сервера, что может привести к раскрытию конфиденциальной информации, например пароля к базе данных, любому пользователю с уровнем Admin в Zabbix.

Attention:
Запуск Zabbix от имени `root`, `bin` или любой другой учетной записи с особыми правами представляет угрозу безопасности.

Домашний каталог (необязательно)

Процессам Zabbix не требуется домашний каталог, поэтому его создание обычно не рекомендуется. Однако, если вам нужна функциональность, требующая домашнего каталога (например, хранение учетных данных MySQL в $HOME/.my.cnf), вы можете создать его с помощью команд, приведенных ниже.

В системах на базе RedHat выполните:

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

В системах на базе Debian выполните:

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

3 Создание базы данных Zabbix

Для демонов Zabbix сервера и прокси, а также веб-интерфейса Zabbix требуется база данных. Она не нужна для запуска Zabbix агента.

SQL скрипты предоставляются для создания схемы базы данных и вставки набора данных. Базе данных Zabbix прокси требуется только схема, тогда как базе данных Zabbix сервера помимо схемы также требуется набор данных.

После создания базы данных Zabbix переходите к следующим шагам компиляции Zabbix.

4 Настройка исходного кода

Для сборки сервера Zabbix, прокси Zabbix или агента Zabbix требуется C99 с расширениями GNU. Эту версию можно явно указать, задав CFLAGS="-std=gnu99":

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

> **Note:**
> При установке из репозитория Git Zabbix сначала необходимо выполнить:
> `./bootstrap.sh`

При настройке исходного кода для сервера Zabbix или прокси необходимо указать тип используемой базы данных. Одновременно с процессом сервера или прокси можно скомпилировать только один тип базы данных.

Чтобы просмотреть все поддерживаемые параметры настройки, выполните следующую команду в распакованном каталоге с исходным кодом Zabbix:

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

Для настройки исходного кода сервера Zabbix и агента можно выполнить команду, подобную следующей:

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

Для настройки исходного кода сервера Zabbix (с PostgreSQL и т. д.) можно выполнить:

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

Для настройки исходного кода прокси Zabbix (с SQLite и т. д.) можно выполнить:

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

Для настройки исходного кода агента Zabbix можно выполнить:

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

или для агента Zabbix 2:

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

> **Note:**
> Для сборки агента Zabbix 2 необходимо установить поддерживаемую версию Go.

Примечания к параметрам компиляции:

- `--enable-agent` - компилирует агент Zabbix, а также утилиты командной строки Zabbix get и Zabbix sender.
- `--with-libcurl` - требуется для мониторинга виртуальных машин, аутентификации SMTP, элементов данных `web.page.*` агента Zabbix и элементов данных запроса телеметрии. См. также: Требования (libcurl).
- `--with-libxml2` - требуется для мониторинга виртуальных машин.

- `--with-libpcre2[=DIR]` - Zabbix всегда компилируется с библиотекой PCRE2; этот параметр позволяет только указать пользовательский путь установки PCRE2.
- `--with-mysql=/path/to/mysql_config` - указывает путь к конфигурации определенной клиентской библиотеки MySQL. Полезно, если установлено несколько версий MySQL или MariaDB.
- `--enable-static` - статически связывает библиотеки (не поддерживается в Solaris). Используйте этот параметр, если планируете распространять скомпилированные двоичные файлы в системах без необходимых библиотек. Не рекомендуется при сборке сервера Zabbix. Для статической сборки сервера требуется статическая версия каждой внешней библиотеки. Скрипт configure автоматически это не проверяет.
- `--with-stacksize=<value>` - задает размер стека каждого потока в килобайтах (например, `--with-stacksize=512`). Это значение можно увеличить, если Zabbix аварийно завершает работу или зависает из-за переполнения стека (например, во время предобработки в системах с низкими ограничениями размера стека потоков по умолчанию).
- `--with-apm` (используйте только с `--enable-proxy`) - требуется для мониторинга производительности приложений (APM) на прокси Zabbix. См. также: Требования (g++, gRPC, protobuf, Abseil).

Attention:
Если `./configure` завершается с ошибкой из-за отсутствующих библиотек или других проблем, проверьте файл `config.log` для получения подробной информации об ошибке.
Например, если отсутствует `libssl`, непосредственное сообщение об ошибке может вводить в заблуждение:
```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```
В этом случае в `config.log` указана фактическая причина:
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

См. также:

- Компиляция Zabbix с поддержкой шифрования
- Известные проблемы компиляции

5 Сборка и установка всего необходимого

Note:
Если установка выполняется из репозитория Zabbix Git, сначала необходимо выполнить:
```
$ make dbschema
```

```
make install
```

Этот шаг следует выполнять от имени пользователя с достаточными правами (обычно `root` или с использованием `sudo`).

При выполнении `make install` по умолчанию будут установлены двоичные файлы демона (zabbix_server, zabbix_agentd, zabbix_proxy) в /usr/local/sbin, а клиентские двоичные файлы (zabbix_get, zabbix_sender) — в /usr/local/bin.

Note:
Чтобы указать другое расположение вместо `/usr/local`, используйте ключ `--prefix` на предыдущем шаге настройки исходных кодов, например `--prefix=/home/zabbix`. В этом случае двоичные файлы демона будут установлены в `<prefix>/sbin`, а утилиты — в `<prefix>/bin`. Страницы man будут установлены в `<prefix>/share`.

6 Просмотр и редактирование файлов конфигурации

- отредактируйте файл конфигурации агента Zabbix (`zabbix_agentd.conf`).

Необходимо настроить этот файл для каждого узла сети, на котором установлен `zabbix_agentd`.

В файле необходимо указать **IP-адрес** сервера Zabbix. Подключения с других узлов сети будут отклоняться.

- отредактируйте файл конфигурации сервера Zabbix (`zabbix_server.conf`).

Необходимо указать имя базы данных, имя пользователя и пароль (если они используются).

Остальные параметры можно оставить со значениями по умолчанию, если у вас небольшая установка (до десяти отслеживаемых узлов сети). Однако следует изменить параметры по умолчанию, если вы хотите максимально повысить производительность сервера Zabbix (или прокси).

- если вы установили прокси Zabbix, отредактируйте файл конфигурации прокси (`zabbix_proxy.conf`).

Необходимо указать IP-адрес сервера и имя узла сети прокси (оно должно быть известно серверу), а также имя базы данных, имя пользователя и пароль (если они используются).

Note:
При использовании SQLite необходимо указать полный путь к файлу базы данных; имя пользователя и пароль базы данных не требуются.

7 Запуск демонов

Запустите zabbix_server на стороне сервера.

```
zabbix_server
```

Note:
Убедитесь, что ваша система позволяет выделить 36 МБ (или немного больше) разделяемой памяти, в противном случае сервер может не запуститься, и вы увидите "Cannot allocate shared memory for <type of cache>." в файле журнала сервера. Это может произойти в FreeBSD, Solaris 8.

Запустите zabbix_agentd на всех контролируемых машинах.

```
zabbix_agentd
```

Note:
Убедитесь, что ваша система позволяет выделить 2 МБ разделяемой памяти, в противном случае агент может не запуститься, и вы увидите "Cannot allocate shared memory for collector." в файле журнала агента. Это может произойти в Solaris 8.

Если вы установили Zabbix прокси, запустите zabbix_proxy.

```
zabbix_proxy
```

Установка веб-интерфейса Zabbix

Копирование файлов PHP

Веб-интерфейс Zabbix написан на PHP, поэтому для его запуска требуется веб-сервер с поддержкой PHP. Установка выполняется простым копированием файлов PHP из каталога ui в каталог HTML-документов веб-сервера.

К распространенным расположениям каталогов HTML-документов для веб-серверов Apache относятся:

- /usr/local/apache2/htdocs (каталог по умолчанию при установке Apache из исходного кода)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Рекомендуется использовать подкаталог вместо корня HTML. Чтобы создать подкаталог и скопировать в него файлы веб-интерфейса Zabbix, выполните следующие команды, заменив <htdocs> на фактический каталог:

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

Если вы планируете использовать любой язык, кроме английского, см. Установка дополнительных языков веб-интерфейса для получения инструкций.

Установка веб-интерфейса

Пожалуйста, смотрите страницу Установка веб-интерфейса для получения информации о мастере установки веб-интерфейса Zabbix.

Установка загружаемых плагинов Zabbix agent 2

Установка загружаемых плагинов Zabbix agent 2 требуется только в том случае, если вы хотите отслеживать цели, не поддерживаемые встроенными плагинами (например, серверы или кластеры MongoDB, PostgreSQL и его форки и т. д.). См. полный список загружаемых плагинов и встроенных плагинов.

Attention:
Перед установкой плагина, пожалуйста, ознакомьтесь с файлом README. Он может содержать особые требования и инструкции по установке.

Чтобы установить из исходных кодов, сначала скачайте и распакуйте архив исходных кодов загружаемого плагина.

Чтобы скомпилировать плагин, перейдите в каталог распакованного плагина и выполните `make`:

```
make
```

Note:
Для сборки загружаемых плагинов Zabbix agent 2 должна быть установлена поддерживаемая версия Go.

Исполняемый файл плагина может находиться в любом месте, если он доступен для загрузки Zabbix agent 2. Укажите путь к двоичному файлу плагина в файле конфигурации плагина, например, в `postgresql.conf` для плагина PostgreSQL:

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

Путь к файлу конфигурации плагина должен быть указан в параметре `Include` файла конфигурации Zabbix agent 2:

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

Перейдите к настройке для получения дополнительных сведений о настройке плагинов.

Загружаемые плагины, предоставляемые Zabbix, используют простые makefile со следующими целями сборки:

- `make` - собрать плагин
- `make clean` - удалить все файлы, созданные при сборке плагина
- `make check` - выполнить самотестирование (требуется реальная цель мониторинга, например база данных PostgreSQL)
- `make style` - проверить стиль кода Go с помощью `golangci-lint`
- `make format` - форматировать код Go с помощью `go fmt`
- `make dist` - создать архив исходных кодов, включая все зависимости

Установка Java gateway

Устанавливать Java gateway требуется только в том случае, если вы хотите отслеживать приложения JMX. Java gateway является легковесным и не требует базы данных.

Чтобы установить из исходных кодов, сначала скачайте и распакуйте архив с исходным кодом.

Чтобы скомпилировать Java gateway, выполните скрипт `./configure` с параметром `--enable-java`. Рекомендуется указать параметр `--prefix`, чтобы задать путь установки, отличный от значения по умолчанию /usr/local, поскольку при установке Java gateway будет создано целое дерево каталогов, а не только один исполняемый файл.

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

Чтобы скомпилировать и упаковать Java gateway в файл JAR, выполните `make`. Обратите внимание, что для этого шага вам понадобятся исполняемые файлы `javac` и `jar` в вашем пути.

```
make
```

Теперь у вас есть файл zabbix-java-gateway-$VERSION.jar в `src/zabbix_java/bin`. Если вам удобно запускать Java gateway из `src/zabbix_java` в каталоге дистрибутива, тогда вы можете перейти к инструкциям по настройке и запуску Java gateway. В противном случае убедитесь, что у вас достаточно прав, и выполните `make install`.

```
make install
```

Перейдите к настройке для получения более подробной информации о настройке и запуске Java gateway.

Установка веб-сервиса Zabbix

Установка веб-сервиса Zabbix требуется только в том случае, если вы хотите использовать запланированные отчеты.

Чтобы установить из исходных кодов, сначала скачайте и распакуйте архив с исходным кодом.

Чтобы скомпилировать веб-сервис Zabbix, выполните скрипт `./configure` с опцией `--enable-webservice`.

Note:
Для сборки веб-сервиса Zabbix должна быть установлена поддерживаемая версия Go.

Запустите zabbix_web_service на машине, где установлен веб-сервис:

```
zabbix_web_service
```

Перейдите к настройке для получения дополнительных сведений о настройке генерации запланированных отчетов.

Установка MCP-сервера Zabbix

Для установки из исходного кода сначала скачайте и распакуйте архив с исходным кодом.

Чтобы скомпилировать MCP-сервер Zabbix, выполните следующую команду в каталоге с исходным кодом MCP-сервера:

```
make build
```

> **Note:**
> Для сборки MCP-сервера Zabbix необходимо установить поддерживаемую версию Go.

Чтобы указать определенную версию в выводе команды `zabbix_mcp_server -V` и в заголовке `User-Agent` запросов к API Zabbix, задайте переменную `VERSION`, например: `make build VERSION=8.0.0`.

Чтобы установить двоичный файл в `/usr/local/sbin`, а пример файла конфигурации - в `/etc/zabbix`, выполните:

```
sudo make install
```

Отредактируйте файл конфигурации `/etc/zabbix/zabbix_mcp_server.conf` и задайте как минимум следующие параметры:

- `FrontendURL` - URL API Zabbix, заканчивающийся на `api_jsonrpc.php`;
- `AllowedIP` - адреса клиентов MCP, которым разрешено подключаться.

Чтобы шифровать подключения от клиентов MCP, задайте `TLSAccept=cert` и укажите `TLSCertFile` и `TLSKeyFile` либо запустите MCP-сервер за доверенным обратным прокси, который обрабатывает TLS.

Если пользователь `zabbix` не существует (например, если на компьютере установлен только MCP-сервер), создайте его, как описано в разделе Создание учетной записи пользователя.

Проверьте файл конфигурации и запустите zabbix_mcp_server:

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

Перейдите к разделу MCP-сервер, чтобы получить дополнительную информацию о настройке сервера и подключении клиентов MCP.

Получение исходного кода Zabbix

Существует несколько способов получить исходный код Zabbix:

- Вы можете скачать выпущенные стабильные версии с официального сайта Zabbix
- Вы можете скачать nightly-сборки со страницы разработчиков на официальном сайте Zabbix
- Вы можете получить последнюю разрабатываемую версию из Git-репозитория исходного кода:
  - Основное расположение полного репозитория: https://git.zabbix.com/scm/zbx/zabbix.git
  - Ветка master и поддерживаемые выпуски также зеркалируются на Github: https://github.com/zabbix/zabbix

Для клонирования репозитория должен быть установлен Git-клиент. Официальный пакет клиента Git для командной строки в дистрибутивах обычно называется **git**. Чтобы установить его, например, в Debian/Ubuntu, выполните:

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

Чтобы получить весь исходный код Zabbix, перейдите в каталог, в котором вы хотите разместить код, и выполните:

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

Проблемы компиляции

Ниже перечислены известные проблемы, связанные с компиляцией Zabbix из исходных кодов. Во всех остальных случаях смотрите страницу Известные проблемы.

Библиотека в нестандартном расположении

Zabbix позволяет указать библиотеку, расположенную в нестандартном месте. В примере ниже Zabbix запустит `curl-config` из указанного нестандартного расположения и использует его вывод, чтобы определить, какую libcurl следует использовать.

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

Это будет работать, если в системе установлена только одна libcurl, но может не сработать, если другая libcurl установлена в стандартном расположении (например, менеджером пакетов). Такая ситуация возникает, когда для Zabbix требуется более новая версия библиотеки, а более старая — для других приложений.

Поэтому указание компонента в нестандартном расположении не всегда будет работать, если тот же компонент также существует в стандартном расположении.

Например, если вы используете более новую libcurl, установленную в `/usr/local`, при этом пакет libcurl по-прежнему установлен, Zabbix может выбрать неправильную библиотеку, и компиляция завершится ошибкой:

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

Здесь функция `curl_easy_header()` недоступна в более старой `/usr/lib/x86_64-linux-gnu/libcurl.so`, но доступна в более новой `/usr/local/lib/libcurl.so`.

Проблема связана с порядком флагов компоновщика, и одним из решений является указание полного пути к библиотеке в переменной LDFLAGS:

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

Обратите внимание на параметр `-Wl,--no-as-needed`, который может потребоваться в некоторых системах (см. также: параметры компоновки по умолчанию в системах на базе Debian-based).

Слишком маленький размер стека на некоторых системах

Если Zabbix аварийно завершается или зависает из-за переполнения стека, вы можете увеличить размер стека для каждого потока, используя параметр `--with-stacksize` при настройке исходных кодов. Эта проблема может возникать в системах с низкими значениями ограничения размера стека потоков по умолчанию, особенно во время предобработки, когда создается несколько потоков.

В следующем примере размер стека для каждого потока устанавливается в 512 КБ:

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

Вы можете проверить ограничения размера стека потоков в системе во время выполнения с помощью команды `ulimit -s` в системах на базе Linux.

Примечания по установке из пакетов

Обзор

Официальные **предварительные** пакеты Zabbix 8.0 доступны на веб-сайте Zabbix, где можно выбрать операционную систему и компонент Zabbix, чтобы сгенерировать инструкции и команды для установки.

Помимо приведённых ниже примечаний по установке пакетов, ознакомьтесь также со страницей Установка и настройка.

Пакеты доступны для следующих дистрибутивов Linux:

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

> **Attention:**
> Некоторые дистрибутивы ОС, в частности дистрибутивы на основе Debian, предоставляют собственные пакеты Zabbix. Эти пакеты **не поддерживаются** Zabbix и могут быть устаревшими или не содержать последние функции и исправления ошибок. Рекомендуется использовать только официальные пакеты из официального репозитория Zabbix. Если ранее вы установили Zabbix из репозитория операционной системы, ознакомьтесь с инструкциями по обновлению пакетов Zabbix из репозиториев ОС.

Пакеты поддерживают базы данных MySQL/PostgreSQL и веб-серверы Apache/Nginx. Обратите внимание, что сервер Zabbix и прокси не могут использовать одну и ту же базу данных; если оба компонента установлены на одном узле сети, используйте разные имена баз данных.

При необходимости отдельные пакеты для агента Zabbix/агента 2, Zabbix get и Zabbix sender доступны в официальном репозитории Zabbix.

В системах на основе Debian или Ubuntu установка дополнительного пакета Zabbix с помощью `apt` также может выбрать обновления для уже установленных пакетов Zabbix из того же репозитория. Чтобы установить пакет без обновления установленного пакета сервера Zabbix или прокси, временно заблокируйте этот пакет перед установкой, а после установки снимите блокировку:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

Zabbix также предоставляет предварительно скомпилированные двоичные файлы агента Zabbix для операционных систем, отличных от Linux; см.:

- Установка агента Windows из MSI
- Установка агента macOS из PKG
- Агент Zabbix в IBM AIX
- Устаревшие двоичные файлы (для старых или менее распространённых систем, таких как HP-UX, NetBSD, Tru64 и более старых версий SLES)

Примечания по установке пакетов

Следующие примечания относятся ко всем системам:

- При использовании PostgreSQL параметр `DBHost=localhost` (или IP-адрес) в конфигурации Zabbix сервера/прокси заставляет PostgreSQL использовать сетевой сокет вместо локального UNIX-сокета; связанные инструкции по настройке см. в разделе Настройка SELinux.
- При использовании TimescaleDB см. дополнительные инструкции по настройке TimescaleDB.
- При установке Java gateway Zabbix (для мониторинга приложений JMX) см. дополнительные инструкции по настройке для систем на базе RHEL и систем на базе Debian.
- При установке MCP-сервера Zabbix установите пакет `zabbix-mcp-server` и обновите файл конфигурации (как минимум параметры `FrontendURL` и `AllowedIP`) перед запуском службы.
- Информацию о запуске агента Zabbix от имени пользователя root см. в разделе Запуск агента от имени пользователя root.
- Мониторинг производительности приложений (APM) на прокси Zabbix поддерживается следующими пакетами:
  - RHEL 10 и его производные: установите пакет APM для прокси Zabbix (например, `zabbix-proxy-mysql-apm` вместо `zabbix-proxy-mysql`); также необходимо установить репозиторий EPEL и включить репозитории пакетов зависимостей CRB.
  - Debian 12 или более поздней версии, Raspberry Pi OS (Raspbian) 12 или более поздней версии, Ubuntu 24.04 или более поздней версии: APM включен в пакеты прокси Zabbix.
  - В других системах можно установить прокси Zabbix из исходного кода с параметром `--with-apm`.

Следующие примечания относятся к RHEL и его производным:

- Если вы включили репозиторий EPEL (например, для пакета APM прокси Zabbix), который также предоставляет пакеты Zabbix, его необходимо исключить из разрешения зависимостей пакетов перед установкой официальных пакетов Zabbix; см. раздел Случайная установка пакетов Zabbix из EPEL.
- Информацию об установке пакетов Zabbix в средах Red Hat UBI см. в разделе Пакеты Zabbix для RHEL в средах Red Hat UBI.
- Для использования элементов данных ICMP ping пакеты `fping` также доступны в официальном репозитории Zabbix.

Настройка SELinux

Zabbix использует межпроцессное взаимодействие на основе сокетов.
На системах, где включен Security-Enhanced Linux (SELinux), может потребоваться добавить правила SELinux, чтобы разрешить Zabbix создавать и использовать UNIX domain sockets в каталоге SocketDir.
Файлы сокетов используются сервером Zabbix (alerter, preprocessing, IPMI) и прокси Zabbix (IPMI), и они присутствуют, пока процесс выполняется.

Если SELinux включен в режиме enforcing, выполните следующие команды, чтобы разрешить обмен данными между веб-интерфейсом Zabbix и сервером:

Для RHEL 7 (и новее), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (и новее):

```
setsebool -P httpd_can_connect_zabbix on
```

Если доступ к базе данных осуществляется по сети (включая `localhost` для PostgreSQL), также разрешите веб-интерфейсу Zabbix подключаться к базе данных:

```
setsebool -P httpd_can_network_connect_db on
```

Для RHEL до 7:

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

После применения настроек SELinux перезапустите Apache:

```
systemctl restart httpd
```

При желании вы можете установить заранее определенный пакет `zabbix-selinux-policy` из официального репозитория Zabbix.
Этот пакет предоставляется для всех поддерживаемых версий ОС, чтобы упростить развертывание Zabbix и предотвратить отключение SELinux пользователями из-за сложности настройки.
 Для версии 10 AlmaLinux, CentOS Stream, RHEL, Rocky Linux и Oracle Linux политика SELinux предоставляется пакетом `zabbix-selinux-policy-main`; пакет

zabbix-selinux-policy имеет зависимость либо от zabbix-selinux-policy-main, либо от selinux-policy-targeted-extra. Пакет zabbix-selinux-policy-main конфликтует с selinux-policy-targeted-extra; поэтому оба пакета не могут быть установлены одновременно.

> **Attention:**
> Для максимальной безопасности рекомендуется задать собственные настройки SELinux.

Пакет zabbix-selinux-policy (zabbix-selinux-policy-main для версии 10 AlmaLinux, CentOS Stream, RHEL, Rocky Linux и Oracle Linux) содержит базовую политику SELinux, которая позволяет Zabbix создавать и использовать сокеты и разрешает HTTPd подключение к PostgreSQL (используется веб-интерфейсом).

Исходный файл zabbix_policy.te содержит следующие правила:

```
module zabbix_policy 1.2;

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

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

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

Пакеты Debuginfo

Пакеты Debuginfo содержат отладочные символы для двоичных файлов Zabbix. Они не требуются для обычной установки или работы, но полезны для расширенного устранения неполадок.

Обратите внимание, что пакеты Debuginfo доступны для следующих версий: Red Hat Enterprise Linux 7 и более ранних (включая производные), а также SUSE Linux Enterprise Server 15 и более ранних (включая производные).

Чтобы включить репозиторий zabbix-debuginfo:

- В RHEL 7 отредактируйте /etc/yum.repos.d/zabbix.repo и установите enabled=1 для раздела zabbix-debuginfo:

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

- В SUSE отредактируйте /etc/zypp/repos.d/zabbix.repo и установите enabled=1 для раздела zabbix-debuginfo:

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

После включения установите пакеты:

- В RHEL установите один пакет с отладочной информацией для всех компонентов Zabbix:

```
dnf install zabbix-debuginfo
```

- В SUSE установите пакеты debuginfo для отдельных компонентов:

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

Включение репозиториев нестабильных выпусков

Ниже приведены инструкции по включению нестабильных репозиториев выпусков Zabbix (по умолчанию отключены), которые используются для кандидатов на выпуск minor-версий Zabbix.

Сначала установите или обновите пакет zabbix-release до последней версии. Чтобы включить пакеты rc в вашей системе, выполните следующее:

Red Hat Enterprise Linux

Откройте файл `/etc/yum.repos.d/zabbix.repo` и установите `enabled=1` для репозитория `zabbix-unstable`.

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

Debian/Ubuntu

Откройте `/etc/apt/sources.list.d/zabbix.list` и раскомментируйте "нестабильный репозиторий Zabbix".

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

SUSE

Откройте файл `/etc/zypp/repos.d/zabbix.repo` и установите `enable=1` для репозитория `zabbix-unstable`.

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

Установка агента Windows из MSI

Обзор

Zabbix агента для Windows можно установить из пакетов установщика Windows MSI (32-бит или 64-бит), доступных для загрузки.

Минимальные требования ОС для MSI установки:

- **для Zabbix агента:** Windows XP (64-бит) или Windows Сервер 2003 ;
- **для Zabbix агента 2:** Windows 10 (32-бит) или Windows Сервер 2016.

32-битный пакет нельзя установить на 64-битную Windows.

Пакеты включают:

- Поддержку TLS (настройка TLS необязательна)
- Утилиты Zabbix get и Zabbix sender (могут быть установлены вместе с Zabbix агентом/агентом 2 или отдельно)

> **Attention:**
> Пакеты Zabbix агента 2 не включают подгружаемые плагины (MongoDB, PostgreSQL, MSSQL), которые нужно загрузить и установить отдельно.

Установку можно выполнить с помощью Мастера настройки (Setup Wizard) или командной строки.

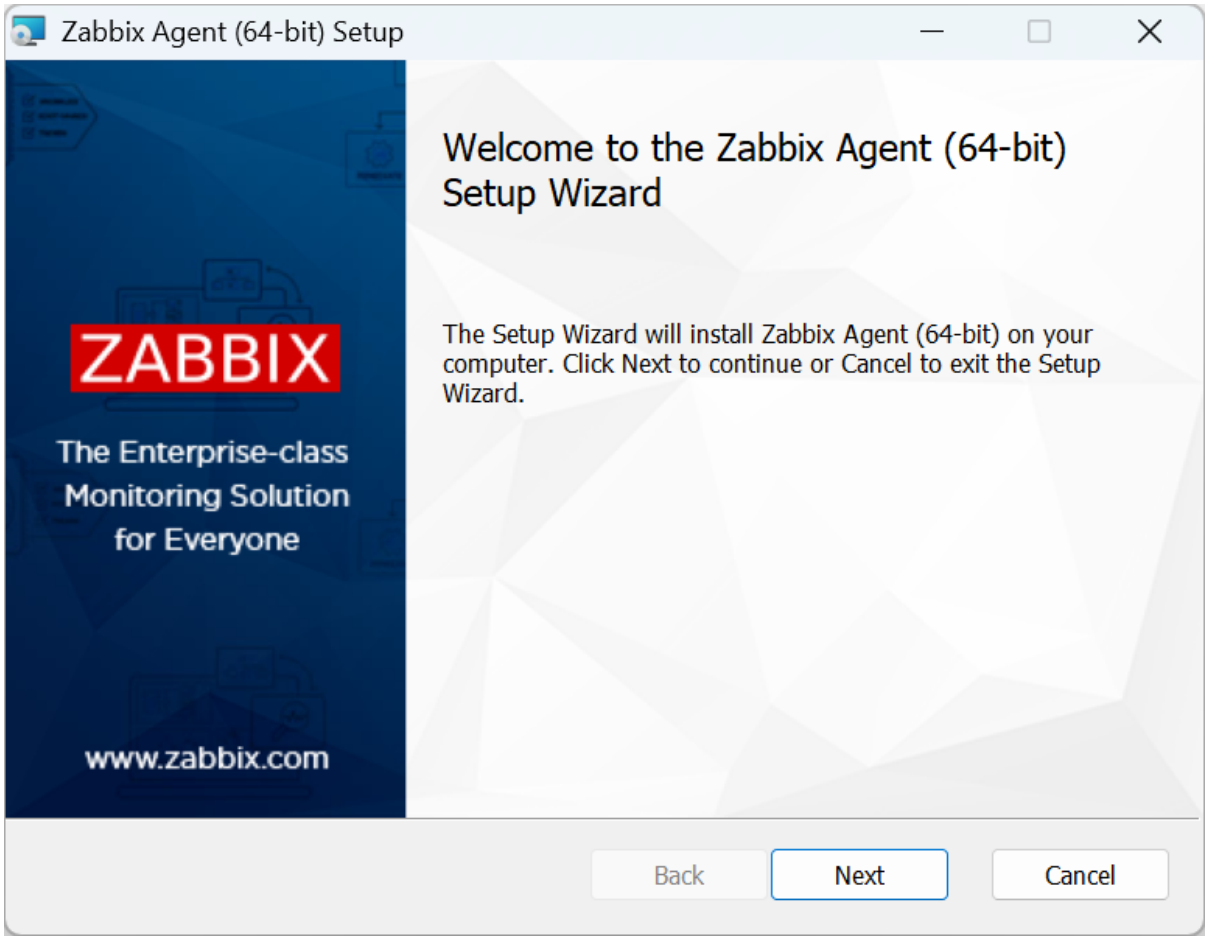
Хотя установка из пакетов установщика MSI полностью поддерживается, рекомендуется установить, как минимум, Microsoft .NET Framework 2 для правильной обработки ошибок.

Attention:
Рекомендуется использовать пути по умолчанию, предоставленные установщиком. Использование пользовательских путей без соответствующих разрешений может поставить под угрозу безопасность установки.

Установка с помощью мастера установки

Следующие шаги установки применимы как к Zabbix agent, так и к Zabbix agent 2.

1. Дважды щелкните загруженный MSI-файл, чтобы начать установку:

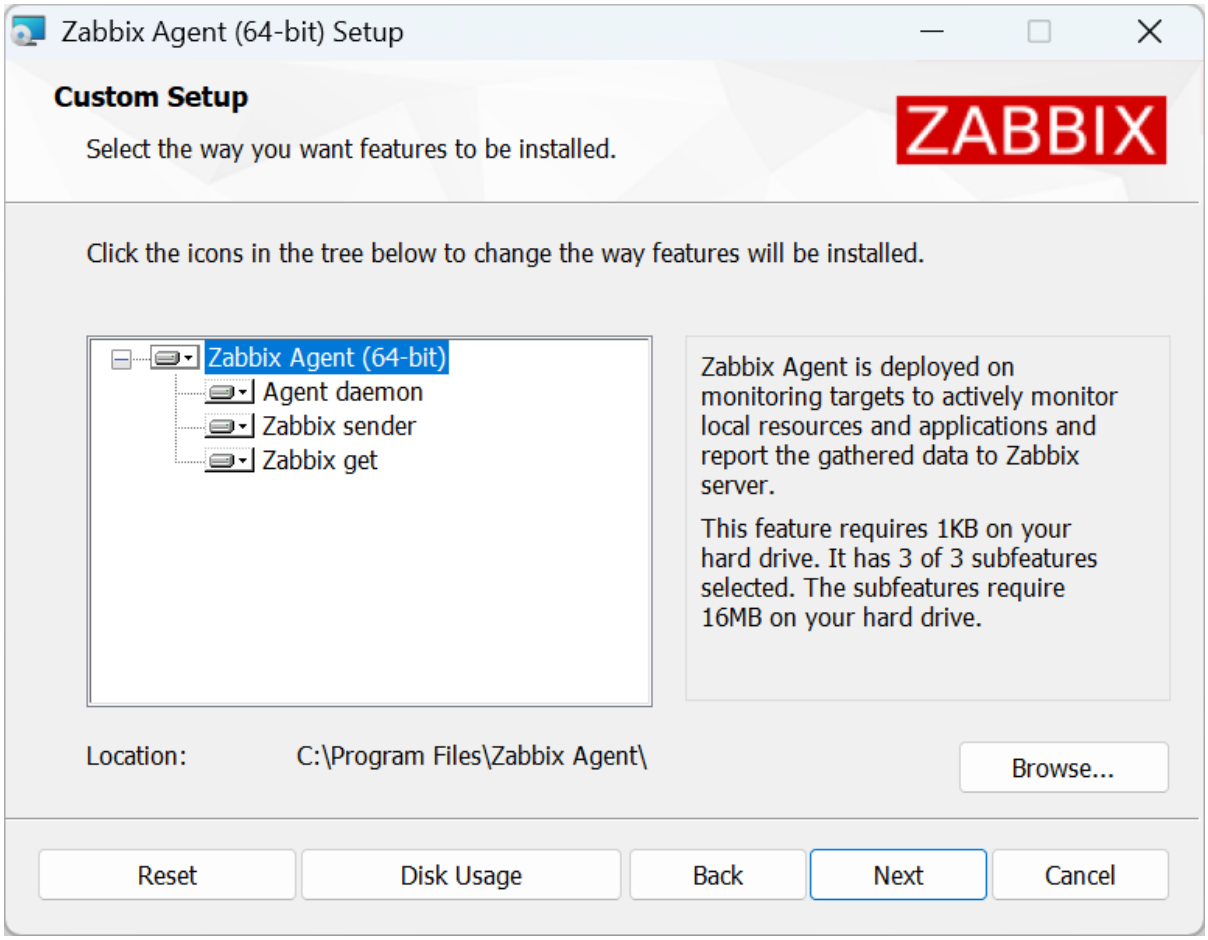


2. Примите лицензионное соглашение с конечным пользователем:



3. Выберите компоненты Zabbix (служба агента, Zabbix sender, Zabbix get), которые необходимо установить:

Attention:
Рекомендуется использовать расположение установки по умолчанию, предлагаемое установщиком. Использование пользовательского расположения без необходимых разрешений может поставить под угрозу безопасность установки.

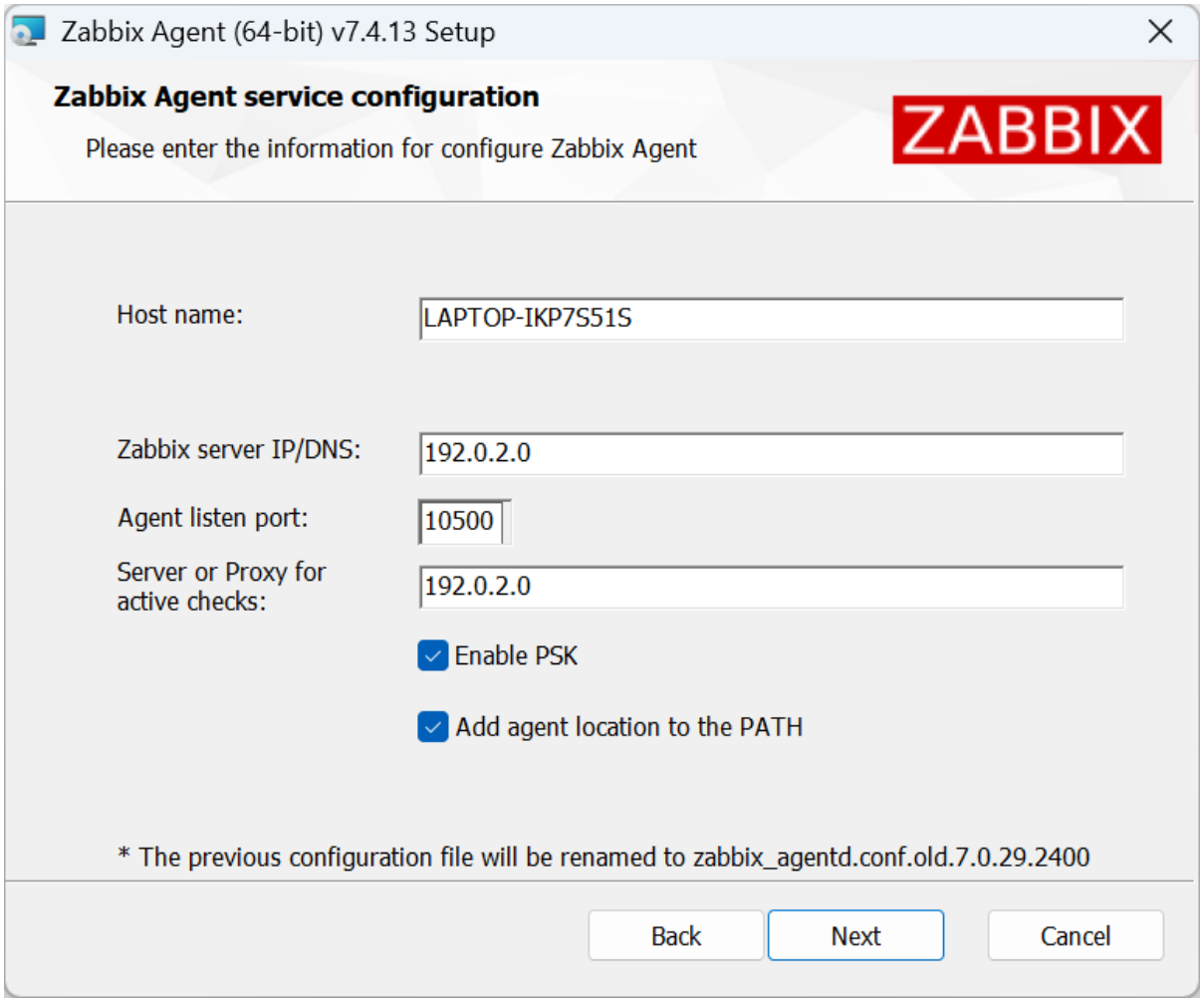


Note:
Если используется пользовательское расположение за пределами каталога `Program Files`, а папка назначения уже существует, установщик не изменит ее разрешения. При новой установке вам будет предложено либо вручную установить соответствующие разрешения, либо продолжить без их изменения.

4. Настройте следующие параметры. Их значения будут указаны в файле конфигурации агента Zabbix:

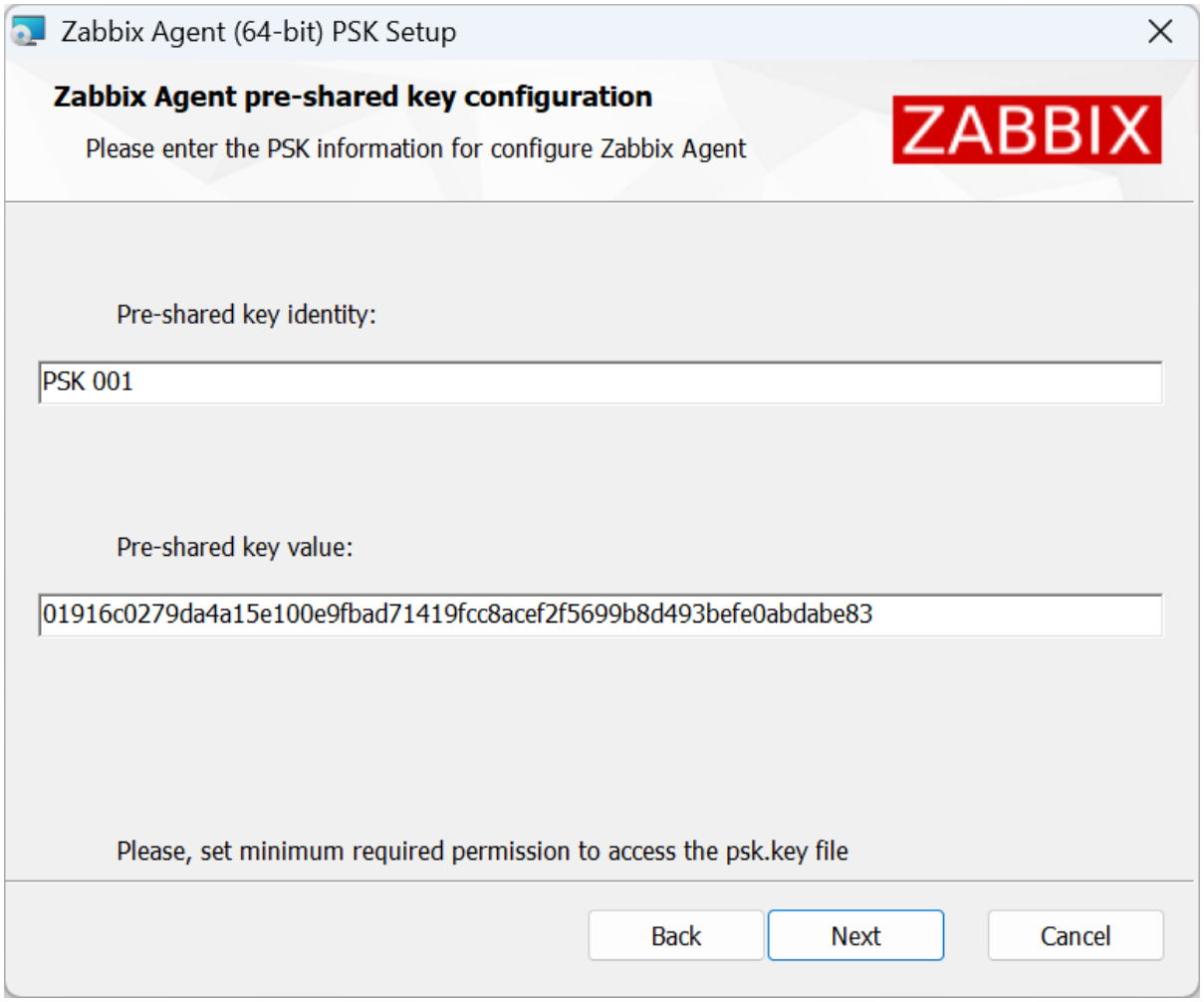
Параметр	Описание
Имя узла сети	Имя узла сети компьютера, на котором устанавливается агент Zabbix. Задает параметр Hostname.
IP-адрес/DNS-сервер Zabbix	Список разделенных запятыми IP-адресов, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix либо прокси Zabbix. Этот параметр является **обязательным**. Задает параметр Server.
Порт прослушивания агента	Агент будет прослушивать этот порт для подключений от сервера. Задает параметр ListenPort.
Сервер или прокси для активных проверок	Адрес сервера/прокси Zabbix или конфигурация кластера, откуда будут получаться активные проверки. Адрес сервера/прокси представляет собой IP-адрес или DNS-имя и необязательный порт, разделенные двоеточием. Задает параметр ServerActive.
Включить PSK	Установите флажок, чтобы включить поддержку TLS с использованием предварительно установленных ключей. Устанавливает для параметров TLSConnect и TLSAccept значение `psk`.
Добавить расположение агента в PATH	Установите флажок, чтобы добавить расположение агента Zabbix в системную переменную PATH.

Note:
Если обнаружен существующий агент Zabbix, будут отображены значения параметров из его файла конфигурации. Кроме того, во время установки существующий файл конфигурации будет переименован, а новый файл конфигурации будет создан.



5. Настройте параметры PSK, если на предыдущем шаге вы установили флажок *Включить PSK*. Эти параметры также будут указаны в файле конфигурации агента Zabbix:

Параметр	Описание
Идентификатор предварительно установленного ключа	Строка идентификатора предварительно установленного ключа. Задает параметр TLSPSKIdentity.
Значение предварительно установленного ключа	Строковое значение предварительно установленного ключа. Создает файл `psk.key`, содержащий ключ, и задает для параметра TLSPSKFile расположение ключа (по умолчанию: `C:\Program Files\Zabbix Agent\psk.key`). **Рекомендуется** ограничить доступ к файлу предварительно установленного ключа, изменив параметры безопасности файла так, чтобы его мог читать только агент Zabbix (или пользователь, от имени которого работает агент).

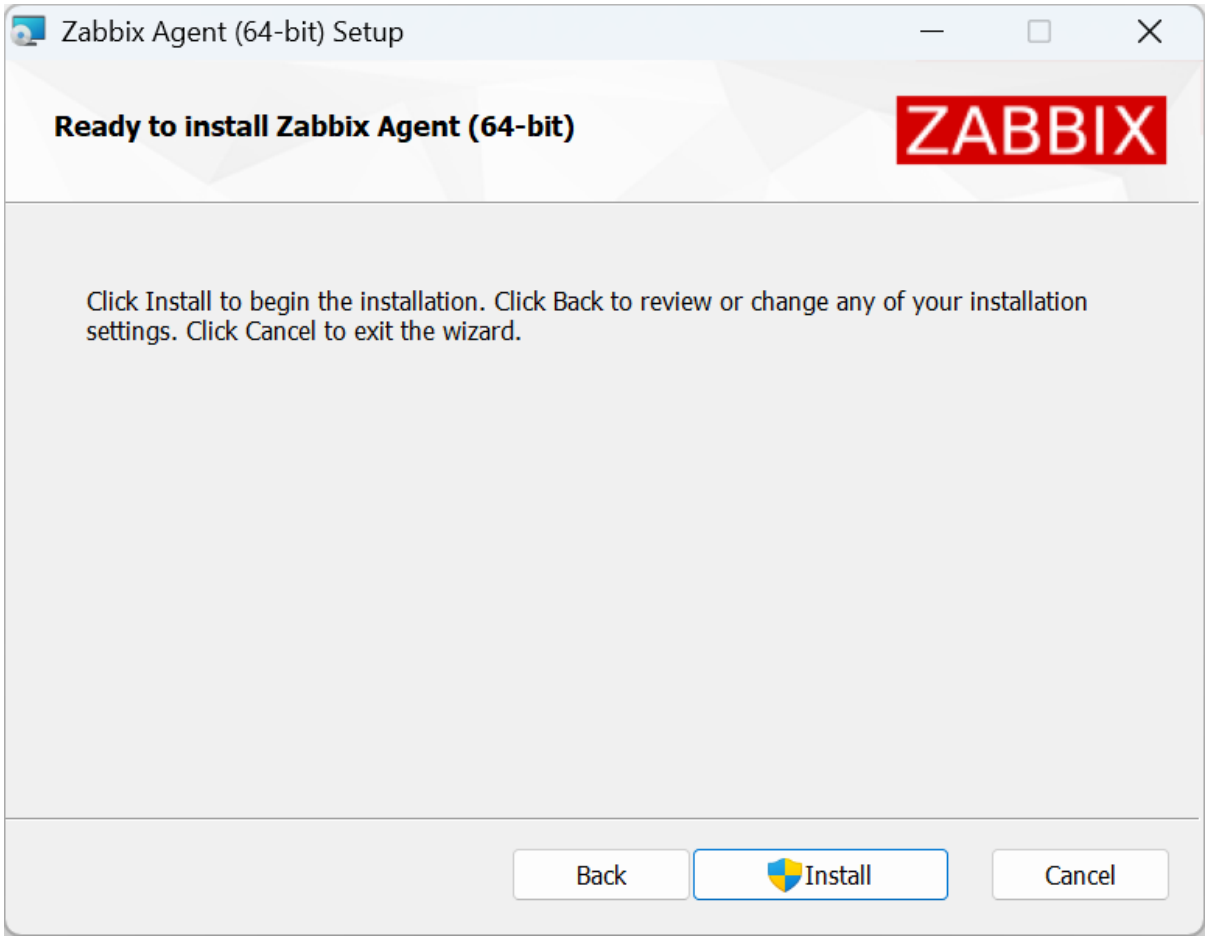


6. Нажмите *Установить*, чтобы начать установку.

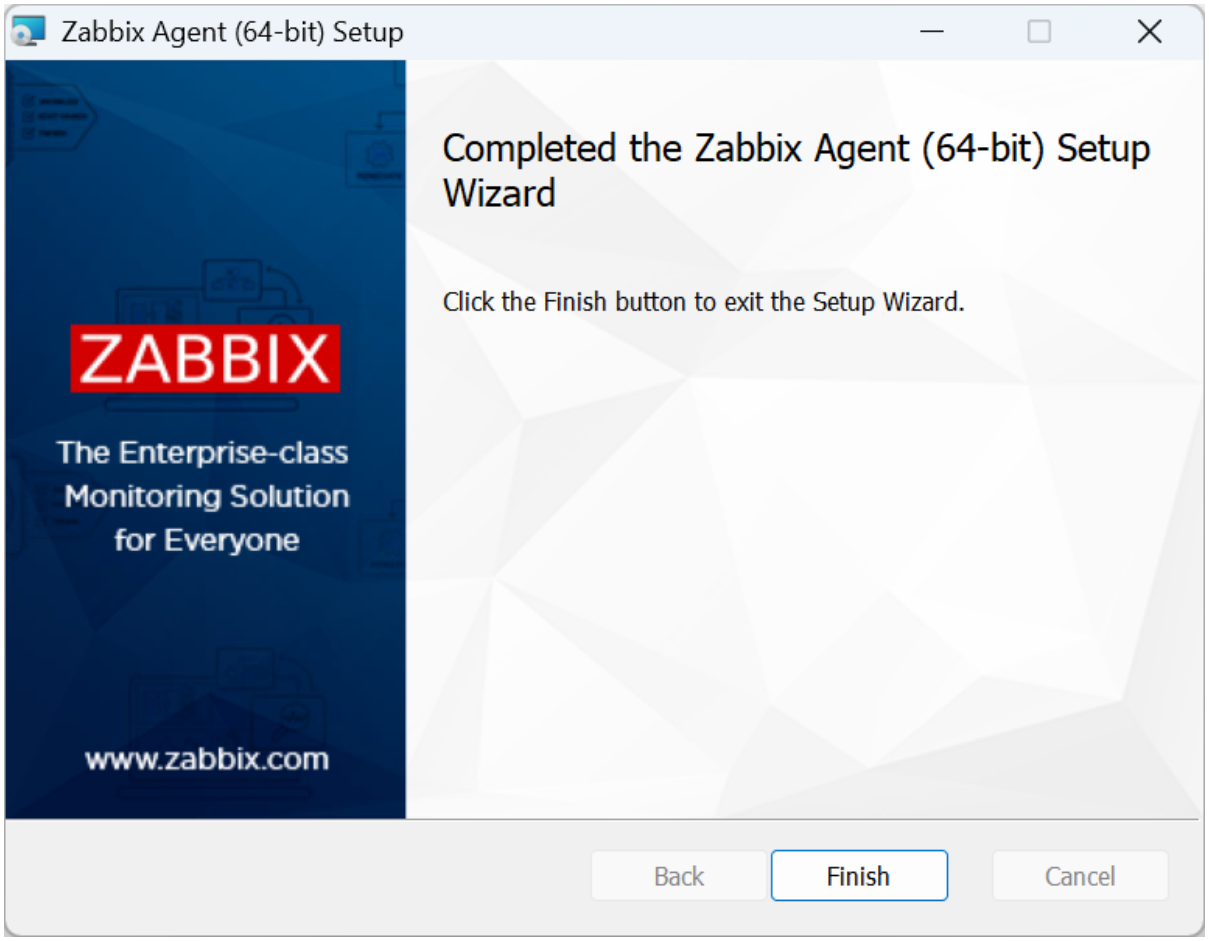
Все выбранные компоненты Zabbix и файл конфигурации агента Zabbix будут установлены в указанное вами расположение (по умолчанию: `C:\Program Files\Zabbix Agent`). То же относится к Zabbix agent 2, за исключением того, что дополнительные файлы конфигурации для его встроенных подключаемых модулей будут установлены в подкаталог `zabbix_agent2.d\plugins.d`.

Кроме того, `zabbix_agentd.exe` (или `zabbix_agent2.exe`) будет настроен как служба Windows с отложенным автоматическим запуском (или автоматическим запуском в версиях Windows до Windows Vista/Server 2008).

Если во время установки работает другая версия агента Zabbix, вам будет предложено либо закрыть приложение и попытаться перезапустить его, либо оставить его открытым; в последнем случае потребуется перезагрузка.



7. Нажмите кнопку *Завершить*, чтобы выйти из мастера установки.



Установка из командной строки

Агент Zabbix можно установить из командной строки, запустив MSI-установщик с помощью `msiexec`. Например:

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

Этот способ позволяет выполнять установку без участия пользователя и использовать параметры для настройки.

Поддерживаемые параметры

Пакеты MSI-установщика Zabbix agent поддерживают следующие параметры как для Zabbix agent, так и для Zabbix agent 2.

Note:
Параметры Zabbix agent/agent2 задаются в файле конфигурации во время установки. Щелкните имя параметра, чтобы просмотреть его подробное описание и примеры настройки на странице Zabbix agent (Windows). Для Zabbix agent 2 см. страницу Zabbix agent 2 (Windows).

Parameter	Description
ADDDEFAULT	Список компонентов, разделенных запятыми, для установки в конфигурации по умолчанию. Дополнительные сведения см. в свойстве ADDDEFAULT. Возможные значения: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Пример: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Список компонентов, разделенных запятыми, для локальной установки. Дополнительные сведения см. в свойстве ADDLOCAL. Возможные значения: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Пример: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Список параметров AllowKey или DenyKey, разделенных точкой с запятой, для ограничения проверок Zabbix agent. При необходимости используйте обратную косую черту для экранирования разделителя (`\;`). Задает параметры AllowKey и DenyKey в файле конфигурации агента. Пример: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Полный путь к файлу конфигурации шаблона для Zabbix agent. Во время установки этот файл станет файлом конфигурации агента. Файл должен содержать как минимум параметры Server и LogFile. Пример: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Используйте `DONOTSTART=1`, чтобы MSI-установщик не запускал службу Zabbix agent.
ENABLEPATH	Используйте `ENABLEPATH=1`, чтобы добавить расположение Zabbix agent в системную переменную PATH.
ENABLEPERSISTENTBUFFER	Только для Zabbix agent 2. Включает использование локального постоянного хранилища для активных элементов данных.
HOSTINTERFACE	Необязательный параметр, определяющий интерфейс узла сети.
HOSTMETADATA	Необязательный параметр, определяющий метаданные узла сети.
HOSTMETADATAITEM	Необязательный параметр, определяющий элемент данных, используемый для получения метаданных узла сети.
HOSTNAME	Необязательный параметр, определяющий имя узла сети.
HOSTNAMEITEM	Необязательный параметр, определяющий элемент данных, используемый для получения имени узла сети.
INCLUDE	Список отдельных файлов, разделенных точкой с запятой, или всех файлов в каталоге, которые нужно включить в файл конфигурации Zabbix agent.
IGNOREALREADYEXISTSFOLDER	Для новых установок используйте `IGNOREALREADYEXISTSFOLDER=1`, чтобы разрешить установку в существующую папку назначения вне каталога `Program Files`. Если этот параметр не указан, установка завершится с ошибкой, если такая папка назначения уже существует.
INSTALLFOLDER	Полный путь к папке, в которую будут установлены компоненты Zabbix и файл конфигурации Zabbix agent. Для Zabbix agent 2 дополнительные файлы конфигурации для встроенных плагинов будут установлены в подпапку `zabbix_agent2.d\plugins.d`. Пример: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Список IP-адресов, разделенных запятыми, на которых агент должен прослушивать подключения.
LISTENPORT	Агент будет прослушивать этот порт для подключений от сервера.
LOGFILE	Имя файла журнала Zabbix agent.
LOGTYPE	Тип вывода журнала.

Parameter	Description
NONMSICONFNAME	Полный путь к пользовательскому файлу конфигурации для Zabbix agent. Во время установки любые допустимые параметры конфигурации агента, присутствующие в этом файле (ограниченные перечисленными в этой таблице), будут записаны в новый файл конфигурации агента. Файл должен содержать как минимум параметр Server. Пример: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Только для Zabbix agent 2. Файл, в котором Zabbix agent 2 должен хранить базу данных SQLite.
PERSISTENTBUFFERPERIOD	Только для Zabbix agent 2. Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси.
SERVER	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix или прокси Zabbix. Этот параметр **обязателен**, кроме случая, когда STARTAGENTS установлен в 0.
SERVERACTIVE	Адрес сервера Zabbix/прокси или конфигурация кластера, откуда получать активные проверки.
SKIP	Используйте `SKIP=fw`, чтобы MSI-установщик не добавлял правило исключения в брандмауэр Windows для Zabbix agent.
SOURCEIP	Исходный IP-адрес для исходящих подключений к серверу Zabbix или прокси Zabbix, а также для установления соединений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т. д.).
STARTAGENTS	Количество предварительно созданных экземпляров zabbix_agentd, обрабатывающих пассивные проверки. Если установлено значение 0, пассивные проверки отключаются, и агент не будет прослушивать ни один TCP-порт.
STARTUPTYPE	Тип запуска службы Zabbix agent. Возможные значения: **automatic** - запускать службу автоматически при старте Windows; **delayed** - *(по умолчанию)* отложить запуск службы после завершения запуска служб, запускаемых автоматически (доступно в Windows Vista/Server 2008 и более поздних версиях); **manual** - запускать службу вручную (пользователем или приложением); **disabled** - отключить службу, чтобы ее нельзя было запустить пользователем или приложением. Пример: `STARTUPTYPE=disabled`
STATUSPORT	Только для Zabbix agent 2. Если задано, агент будет прослушивать этот порт для HTTP-запросов состояния (http://localhost:<port>/status).
TIMEOUT	Указывает, сколько времени ожидать (в секундах) установления соединения и обмена данными с прокси Zabbix или сервером.
TLSACCEPT	Входящие подключения, которые следует принимать (используется для пассивных проверок). Если установлено значение `psk`, то TLSCONNECT также будет установлен в `psk` (если не указано иное).
TLSCAFILE	Полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла.
TLSCERTFILE	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов.
TLSCONNECT	Способ подключения агента к серверу Zabbix или прокси (используется для активных проверок). Если установлено значение `psk`, то TLSACCEPT также будет установлен в `psk` (если не указано иное).
TLSCRLFILE	Полный путь к файлу, содержащему отозванные сертификаты.
TLSKEYFILE	Полный путь к файлу, содержащему закрытый ключ Zabbix agent.
TLSPSKFILE	Полный путь к файлу, содержащему предварительно общий ключ Zabbix agent. Если заданы и TLSPSKFILE, и TLSPSKVALUE, значение TLSPSKVALUE будет записано в файл, указанный в TLSPSKFILE. **Рекомендуется** ограничить доступ к файлу предварительно общего ключа, настроив параметры безопасности файла так, чтобы его мог читать только Zabbix agent (или пользователь, от имени которого запущен агент).
TLSPSKIDENTITY	Строка идентификатора предварительно общего ключа.
TLSPSKVALUE	Строковое значение предварительно общего ключа. Если заданы и TLSPSKFILE, и TLSPSKVALUE, значение TLSPSKVALUE будет записано в файл, указанный в TLSPSKFILE. Пример: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	Допустимый издатель сертификата сервера (прокси).
TLSSERVERCERTSUBJECT	Допустимый субъект сертификата сервера (прокси).
UNSAFEUSERPARAMETERS	Разрешить передачу всех символов в аргументах пользовательских параметров.

Примеры

Следующий пример устанавливает агент Zabbix с пользовательской конфигурацией. Он также включает поддержку TLS с использованием предварительно общих ключей.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

Следующий пример устанавливает более новую версию агента Zabbix и использует **файл конфигурации шаблона** (`CONF="C:\agent-template.conf"`). Во время установки этот файл станет файлом конфигурации агента. Чтобы унаследовать параметры из старого файла конфигурации, используйте заполнители параметров (например, `[AllowDenyKey]`).

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

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

В качестве альтернативы можно использовать **пользовательский файл конфигурации** (`NONMSICONFNAME="C:\agent-custom.co`). Во время установки любые допустимые параметры конфигурации агента, присутствующие в этом файле (ограниченные перечисленными в таблице выше), будут записаны в новый созданный файл конфигурации агента. Чтобы сохранить существующую конфигурацию агента, определите параметры, которые нужно сохранить.

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

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

Загружаемые плагины Zabbix agent 2

Загружаемые плагины Zabbix agent 2 loadable plugins можно установить в Windows с помощью 64-разрядных MSI-пакетов установщика, доступных для загрузки.

Минимальные требования к ОС для установки MSI: Windows 10 (64-bit) или Windows Server 2016.

> **Attention:**
> Перед установкой плагина, пожалуйста, ознакомьтесь с его файлом README. Он может содержать особые требования и инструкции по установке.

Как и Zabbix agent/agent2, загружаемые плагины можно установить с помощью мастера установки или из командной строки.

Установка с помощью мастера настройки

1. Дважды щёлкните загруженный файл MSI, чтобы начать установку.

2. Примите лицензионное соглашение конечного пользователя.

3. Выберите для установки подгружаемый плагин Zabbix агента 2.

Attention:
Рекомендуется использовать пути по умолчанию, предоставленные установщиком. Использование пользовательских путей без соответствующих разрешений может поставить под угрозу безопасность установки

4. Нажмите *Install*, чтобы начать установку.

Все выбранные подгружаемые модули Zabbix агента 2 будут установлены в указанное место (по умолчанию: `C:\Program Files\Zabbix Agent 2`), а из конфигурационные файлы будут установлены в подпапку `zabbix_agent2.d`.

5. Нажмите кнопку *Finish*, чтобы выйти из мастера установки.

Установка из командной строки

Подключаемые плагины Zabbix agent 2 можно установить из командной строки, запустив MSI-установщик с помощью `msiexec`. Например:

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

Пакеты MSI-установщика подключаемых плагинов Zabbix agent 2 поддерживают следующие параметры.

Parameter	Description
ADDDEFAULT	Список компонентов, разделенных запятыми, которые будут установлены в конфигурации по умолчанию. Дополнительные сведения см. в свойстве ADDDEFAULT. Возможные значения: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Пример: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Список компонентов, разделенных запятыми, которые будут установлены локально. Дополнительные сведения см. в свойстве ADDLOCAL. Возможные значения: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Пример: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Полный путь к папке, в которую будут установлены компоненты Zabbix, а их файлы конфигурации — в подпапку `zabbix_agent2.d`. Пример: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Сборка Zabbix агент 2 в Windows

Обзор

На этой странице показано, как собрать агент Zabbix 2 из исходного кода в Windows 10 (64- или 32-разрядной версии).

На 64-разрядной платформе можно собрать как 32-разрядную, так и 64-разрядную версию, но на 32-разрядной платформе можно собрать только 32-разрядную версию.

Для сборки агента Zabbix 2 требуются:

- инструменты сборки MinGW
- язык программирования Go
- OpenSSL (для функций шифрования в Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; для функций сопоставления с шаблонами регулярных выражений в Zabbix)

Собрать агент Zabbix 2 можно одним из следующих способов:

- С использованием vcpkg - автоматизированный подход, упрощающий управление зависимостями с помощью менеджера пакетов C++.
- Сборка вручную - ручной подход, требующий установки всех зависимостей перед компиляцией агента.

Attention:
Перед началом процесса сборки обратите внимание на следующее:

- Для выполнения команд используйте командную строку, запущенную пользователем с достаточными правами для записи в защищенные папки. Однако при установке OpenSSL и PCRE2 используйте терминал MSYS2.
- Рекомендуется создать рабочий каталог `C:\Zabbix` для всех исходных файлов и папок сборки. Однако скомпилированные компоненты следует устанавливать в `C:\Zabbix\x64` (или в `C:\Zabbix\x86` для 32-разрядных сборок).

Сборка агента Zabbix 2 с помощью vcpkg

В этом разделе приведены инструкции по сборке агента Zabbix с помощью vcpkg — менеджера пакетов, упрощающего управление зависимостями и интеграцию с проектами C++.

1. Скачайте и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Разработка классических приложений на C++*, которая включает менеджер пакетов vcpkg.

2. Скачайте и установите Go (доступен в виде установщика MSI). Во время установки укажите `C:\Zabbix\Go` в качестве папки установки.

3. Скачайте дистрибутив MinGW, использующий библиотеку среды выполнения Microsoft Visual C; например:

- Для 64-разрядных сборок: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Для 32-разрядных сборок: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

> **Attention:**
> Версия MinGW должна соответствовать минимальным требованиям.

Затем распакуйте его в `C:\Zabbix\mingw64` (или в `C:\Zabbix\mingw32` для 32-разрядных сборок).

4. Инициализируйте vcpkg и установите зависимости, необходимые для сборки агента Zabbix 2 (обратите внимание, что это может занять некоторое время):

```
cd C:\Zabbix

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

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

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

5. Скачайте архив с исходным кодом Zabbix и распакуйте его в `C:\Zabbix\zabbix-8.0.0`.

6. Перейдите в каталог сборки Zabbix (`C:\Zabbix\zabbix-8.0.0\build\mingw`) и создайте следующий скрипт `build.bat`:

- Для 64-разрядных сборок:

```
::      MinGW   Go              `PATH`          :
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

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

::                          Crypt32:
SET CGO_LDFLAGS="-lCrypt32"

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

- Для 32-разрядных сборок:

```
::      MinGW   Go              `PATH`          :
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%
```

```
::               vcpkg:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

::                          Crypt32:
SET CGO_LDFLAGS="-lCrypt32"

::               :
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Скомпилируйте агент Zabbix 2, выполнив скрипт:

```
build.bat
```

После компиляции двоичный файл агента Zabbix 2 будет находиться в C:\Zabbix\zabbix-8.0.0\bin\win64 (для 64-разрядных сборок) или в C:\Zabbix\zabbix-8.0.0\bin\win32 (для 32-разрядных сборок). Файлы конфигурации агента Zabbix 2 находятся в C:\Zabbix\zabbix-8.0.0\src\go\conf.

Чтобы запустить агент, скопируйте zabbix_agent2.exe и его файлы конфигурации в отдельную папку (например, C:\Zabbix\agent2), а затем запустите агент:

```
mkdir C:\Zabbix\agent2

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

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

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

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

При необходимости продолжите работу, скомпилировав подключаемые плагины агента Zabbix 2.

Компиляция загружаемых плагинов Zabbix agent 2

1. Скачайте исходный код плагина Zabbix, соответствующий версии вашего Zabbix agent 2 (например, zabbix-agent2-plugin-post
и распакуйте его в C:\Zabbix.

> **Attention:**
> Перед установкой плагина обязательно ознакомьтесь с файлом README. Он может содержать особые требования и инструкции по установке.

2. Перейдите в распакованный каталог плагина и скомпилируйте плагин:

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

####    64-         :
mingw32-make ARCH=AMD64

####    32-         :
mingw32-make ARCH=x86
```

После компиляции двоичный файл плагина Zabbix agent 2 и его файл конфигурации будут находиться в том же каталоге плагина.

Исполняемый файл плагина может быть размещен в любом месте, если он может быть загружен Zabbix agent 2. Укажите путь к двоичному файлу плагина в файле конфигурации плагина, например в postgresql.conf для плагина PostgreSQL:

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

Путь к файлу конфигурации плагина должен быть указан в параметре Include файла конфигурации Zabbix agent 2:

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

Перейдите к разделу настройка для получения дополнительных сведений о настройке плагинов.

Сборка Zabbix агент 2 вручную

> **Attention:**
> Этот метод сборки Zabbix агент 2 подходит для пользователей, которым требуется полный контроль над средой сборки, или которые работают в ограниченной среде, где использование vcpkg невозможно.

В этом разделе содержатся инструкции по ручной сборке Zabbix агент 2, включая установку необходимых инструментов сборки и зависимостей, а затем компиляцию агента.

Настройка инструментов сборки

1. Скачайте и установите MSYS2 (доступен в виде MSI-установщика). Во время установки обязательно укажите `C:\Zabbix\msys64` в качестве папки установки.

2. Скачайте и установите Go (доступен в виде MSI-установщика; см. текущие поддерживаемые версии Go). Во время установки обязательно укажите `C:\Zabbix\Go` в качестве папки установки.

3. Скачайте дистрибутив MinGW, который использует библиотеку времени выполнения Microsoft Visual C; например:

- Для 64-разрядных сборок: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Для 32-разрядных сборок: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

> **Attention:**
> Версия MinGW должна соответствовать минимальным требованиям.

Затем распакуйте его в `C:\Zabbix\mingw64` (или `C:\Zabbix\mingw32` для 32-разрядных сборок).

Установка OpenSSL

> **Note:**
> Чтобы скомпилировать агент Zabbix без поддержки TLS, перейдите к разделу Установка PCRE2.

1. Откройте терминал MSYS2 MSYS с правами администратора и выполните следующие команды:

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

2. Загрузите архив с исходным кодом OpenSSL и распакуйте его в `C:\Zabbix\openssl-3.5.0`.

3. Перейдите в распакованный каталог OpenSSL и создайте следующий скрипт `build.sh`:

- Для 64-разрядных сборок:

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

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

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

make
make install
```

- Для 32-разрядных сборок:

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

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

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

```
make
make install
```

Attention:
Обязательно запретите пользователям без прав администратора доступ на запись в каталог C:\Zabbix\x64\OpenSSL-Win64-350-static. В противном случае агент будет загружать настройки SSL из пути, который могут изменять пользователи без соответствующих привилегий, что создает потенциальную уязвимость системы безопасности.

- Параметр no-shared делает статические библиотеки OpenSSL libcrypto.lib и libssl.lib самодостаточными, поэтому двоичные файлы Zabbix включают OpenSSL без необходимости во внешних DLL. Это означает, что двоичные файлы Zabbix можно копировать на другие компьютеры под управлением Windows без библиотек OpenSSL; однако после выпуска новой версии OpenSSL с исправлениями ошибок агент Zabbix потребуется скомпилировать повторно.
- Без параметра no-shared Zabbix использует DLL OpenSSL во время выполнения. Это означает, что обновление OpenSSL может не потребовать повторной компиляции агента Zabbix; однако при копировании агента на другие компьютеры необходимо также скопировать DLL OpenSSL.

Дополнительную информацию о других параметрах конфигурации OpenSSL см. в документации OpenSSL.

4. Настройте и установите OpenSSL, выполнив скрипт (обратите внимание, что это может занять некоторое время):

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

Установка PCRE2

1. Загрузите архив с исходным кодом PCRE2 и распакуйте его в C:\Zabbix\pcre2-10.45.

2. Откройте терминал MSYS2 MSYS с правами администратора. Затем создайте каталог build в распакованном каталоге PCRE2 и перейдите в него:

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

3. Настройте PCRE2:

```
####    64-       :
export PATH="/c/Zabbix/mingw64/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX="/c/Zabbix/x64/PCRE2" ..

####    32-       :
export PATH="/c/Zabbix/mingw32/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE
```

Note:
Если возникают какие-либо ошибки, перед повторной попыткой сборки с помощью CMake рекомендуется удалить кэш CMake. Файл кэша (CMakeCachecache.txt) может находиться в каталоге build распакованного каталога PCRE2.

4. Установите PCRE2:

```
make install
```

Компиляция агента Zabbix 2

1. Загрузите архив с исходным кодом Zabbix и распакуйте его в C:\Zabbix\zabbix-8.0.0.

Если необходимо создать архив с исходным кодом из репозитория исходного кода (например, чтобы применить пользовательские исправления или выполнить сборку из последней версии исходного кода), выполните следующие команды на компьютере с **Linux**, на котором установлен Go (требуется для настройки агента Zabbix 2):

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Будет создан архив с исходным кодом, который затем можно скопировать на компьютер с Windows.

2. Откройте командную строку с правами администратора. Затем перейдите в каталог сборки Zabbix и скомпилируйте агент Zabbix; убедитесь, что каталоги установки OpenSSL и PCRE2 указаны правильно:

- Для 64-разрядных сборок:

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

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

####         TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Для 32-разрядных сборок:

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

####         TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

####         TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

После компиляции двоичный файл агента Zabbix 2 будет находиться в `C:\Zabbix\zabbix-8.0.0\bin\win64` (или в `C:\Zabbix\zabbix-8.0.0\bin\win32` для 32-разрядных сборок). Файлы конфигурации агента Zabbix 2 находятся в `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Чтобы запустить агент, скопируйте двоичный файл `zabbix_agent2.exe` и его файлы конфигурации в отдельный каталог (например, `C:\Zabbix\agent2`), а затем запустите агент:

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

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

При необходимости продолжите компиляцией загружаемых плагинов агента Zabbix 2.

Сборка агента Zabbix в Windows

Обзор

На этой странице показано, как собрать агент Zabbix из исходного кода в Windows 10 (64-разрядная версия).

Эти инструкции применимы к версиям Windows, поддерживающим Visual Studio 2022.

Для сборки агента Zabbix требуются:

- компилятор C (входит в состав Build Tools for Visual Studio 2022);
- OpenSSL (для функций шифрования в Zabbix);
- PCRE2 (Perl Compatible Regular Expressions; для сопоставления с шаблонами регулярных выражений в Zabbix).

Собрать агент Zabbix можно одним из следующих способов:

- С использованием vcpkg - автоматизированный подход, упрощающий управление зависимостями с помощью менеджера пакетов C++;
- Сборка вручную - ручной подход, требующий установки всех зависимостей перед компиляцией агента.

В зависимости от ваших потребностей в мониторинге могут потребоваться дополнительные библиотеки. Дополнительную информацию см. в разделе Требования.

Attention:
Перед началом процесса сборки обратите внимание на следующее:

- Для выполнения команд используйте "x64 Native Tools Command Prompt" (входит в состав Build Tools for Visual Studio 2022), запущенную пользователем с достаточными правами для записи в защищенные папки.
- Рекомендуется создать рабочий каталог `C:\Zabbix` для всех исходных файлов и папок сборки. Однако скомпилированные компоненты следует устанавливать в `C:\Program Files\Zabbix\x64`.

Сборка агента Zabbix с помощью vcpkg

В этом разделе приведены инструкции по сборке агента Zabbix с помощью vcpkg — менеджера пакетов, упрощающего управление зависимостями и интеграцию с проектами C++.

1. Скачайте и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Разработка классических приложений на C++*, которая включает инструменты, необходимые для сборки агента с помощью vcpkg:

- Компилятор C (Microsoft Visual C++)
- Средство командной строки NMake
- Менеджер пакетов vcpkg
- "x64 Native Tools Command Prompt"

2. Откройте `cmd.exe` с правами администратора, инициализируйте vcpkg и установите зависимости, необходимые для сборки агента Zabbix (обратите внимание, что это может занять некоторое время):

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

3. Скачайте архив с исходным кодом Zabbix и распакуйте его в `C:\Zabbix\zabbix-8.0.0`.

4. Перейдите в каталог сборки Zabbix (`C:\Zabbix\zabbix-8.0.0\build\win32\project`) и создайте следующий скрипт `build.bat`; убедитесь, что каталоги установки OpenSSL и PCRE2 указаны правильно:

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

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

5. Откройте "x64 Native Tools Command Prompt" и скомпилируйте агент Zabbix, выполнив скрипт:

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

Если вы открыли другую командную строку, сначала инициализируйте среду сборки Visual Studio:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Заключайте пути, содержащие пробелы, в кавычки.

После компиляции двоичные файлы компонентов Zabbix будут находиться в `C:\Zabbix\zabbix-8.0.0\bin\win64`. Файл конфигурации агента Zabbix находится в `C:\Zabbix\zabbix-8.0.0\conf`.

Чтобы запустить агент, скопируйте `zabbix_agentd.exe` и его файл конфигурации в отдельный каталог (например, `C:\Zabbix\agent`), а затем запустите агент:

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

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

Сборка Zabbix агента вручную

Attention:
Этот метод сборки агента Zabbix подходит пользователям, которым требуется полный контроль над средой сборки или которые находятся в ограниченной среде, где использование vcpkg невозможно.

В этом разделе содержатся инструкции по ручной сборке агента Zabbix, включая установку необходимых инструментов сборки и зависимостей (Perl, OpenSSL, PCRE2), а также компиляцию агента.

Установка инструментов сборки

1. Скачайте и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Desktop development with C++*, которая включает инструменты, необходимые для ручной сборки агента:

- Компилятор C (Microsoft Visual C++)
- Средство командной строки NMake
- "x64 Native Tools Command Prompt"

Установка OpenSSL

Note:
Чтобы скомпилировать агент Zabbix без поддержки TLS, перейдите к разделу Установка PCRE2.

1. Скачайте и установите Strawberry Perl (доступен в виде установщика MSI). Во время установки укажите `C:\Zabbix\Strawberry` в качестве папки установки.

2. Установите модуль Perl `Text::Template`:

```
cpanm Text::Template
```

3. Убедитесь, что Netwide Assembler (NASM; требуется для компиляции OpenSSL) был скомпилирован во время установки Strawberry Perl:

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

Если NASM не скомпилирован, установите его вручную. Дополнительную информацию см. в документации NASM.

4. Скачайте архив с исходным кодом OpenSSL и распакуйте его в `C:\Zabbix\openssl-3.5.0`.

5. Перейдите в распакованный каталог и настройте OpenSSL, например:

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

Attention:
Если при компиляции агента Zabbix в Windows вы выбрали пользовательский каталог для OpenSSL (например, `C:\zabbix` или `C:\openssl-64bit`), обязательно запретите доступ на запись к этому каталогу для пользователей, не имеющих прав администратора. В противном случае агент будет загружать настройки SSL из пути, который могут изменять непривилегированные пользователи, что может привести к уязвимости системы безопасности.

- Параметр `no-shared` делает статические библиотеки OpenSSL libcrypto.lib и libssl.lib самодостаточными, поэтому двоичные файлы Zabbix включают OpenSSL и не требуют внешних DLL. Это означает, что двоичные файлы Zabbix можно копировать на другие компьютеры с Windows без библиотек OpenSSL; однако после выпуска новой версии OpenSSL с исправлениями ошибок агент Zabbix потребуется скомпилировать заново.
- Без параметра `no-shared` Zabbix использует DLL OpenSSL во время выполнения. Это означает, что для обновления OpenSSL может не потребоваться повторная компиляция агента Zabbix; однако при копировании агента на другие компьютеры необходимо также скопировать DLL OpenSSL.

Дополнительную информацию о других параметрах настройки OpenSSL см. в документации OpenSSL.

6. Скомпилируйте OpenSSL и запустите тесты (обратите внимание, что это может занять некоторое время):

Attention:
Запускайте тесты без прав администратора; в противном случае это может привести к неожиданным результатам или рискам для безопасности. Если некоторые тесты завершатся с ошибкой, обратитесь к документации OpenSSL для устранения неполадок.

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

7. Установите OpenSSL:

```
nmake install
```

Чтобы установить только программные компоненты (библиотеки и файлы заголовков без документации), можно использовать `nmake install_sw`.

Установка PCRE2

1. Скачайте и установите [CMake] (https://cmake.org/download/) (доступен в виде MSI-установщика).

Во время установки обязательно укажите `C:\Zabbix\CMake` в качестве папки установки и выберите опцию *Add CMake to the PATH environment variable (Добавить CMake в переменную среды PATH)*.

2. Скачайте [архив исходного кода PCRE2] (https://github.com/PCRE2Project/pcre2/releases/latest) и распакуйте его в `C:\Zabbix\pcre2-10.45`.

3. Создайте каталог `build` в распакованном каталоге PCRE2 и перейдите в него:

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

4. Настройте PCRE2:

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

Note:
При возникновении ошибок рекомендуется удалить кэш CMake перед повторной попыткой сборки CMake. Кэш (`CMakeCachecache.txt`) находится в каталоге build извлечённого каталога PCRE2.

5. Соберите PCRE2 с помощью NMake:

```
nmake
```

6. Установите PCRE2:

```
cmake --install .
```

Компиляция агента Zabbix

1. Скачайте архив с исходным кодом Zabbix и распакуйте его в `C:\Zabbix\zabbix-8.0.0`.

Если необходимо создать архив с исходным кодом из репозитория исходного кода (например, чтобы применить пользовательские исправления или выполнить сборку из последней версии исходного кода), выполните следующие команды на компьютере под управлением **Linux**:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Будет создан архив с исходным кодом, который затем можно скопировать на компьютер под управлением Windows.

2. Перейдите в каталог сборки Zabbix, откройте «x64 Native Tools Command Prompt» и скомпилируйте агент Zabbix (или другие компоненты); убедитесь, что каталоги установки OpenSSL и PCRE2 указаны правильно:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64

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

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

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

Заключайте пути, содержащие пробелы, в кавычки.

После компиляции двоичные файлы компонентов Zabbix будут находиться в `C:\Zabbix\zabbix-8.0.0\bin\win64`. Файл конфигурации агента Zabbix находится в `C:\Zabbix\zabbix-8.0.0\conf`.

Чтобы запустить агент, скопируйте `zabbix_agentd.exe` и его файл конфигурации в отдельный каталог (например, `C:\Zabbix\agent`), а затем запустите агент:

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

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

Установка агента для macOS из PKG

Обзор

Агент Zabbix можно установить в macOS с помощью установочных пакетов PKG, доступных для загрузки.

Пакеты агента Zabbix доступны с шифрованием и без него.

Установка агента

Агент можно установить с помощью графического пользовательского интерфейса или из командной строки, например:

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

Убедитесь, что в команде используется правильная версия пакета Zabbix. Она должна совпадать с именем загруженного пакета.

Запуск агента

Агент автоматически запустится после установки или перезагрузки.

При необходимости вы можете отредактировать файл конфигурации в `/usr/local/etc/zabbix/zabbix_agentd.conf`.

Чтобы запустить агента вручную, вы можете выполнить:

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

Для остановки агента вручную:

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

В процессе обновления существующий файл конфигурации не будет перезаписан. Вместо этого будет создан новый файл `zabbix_agentd.conf.NEW`, который можно просмотреть и затем обновить существующий файл конфигурации, если необходимо. Не забудьте перезапустить агента после ручных изменений в файле конфигурации.

Устранение неполадок и удаление агента

В этом разделе перечислены некоторые полезные команды, которые можно использовать для поиска неполадок и удаления установленного Zabbix агента.

Просмотр, запущен ли Zabbix агент:

```
ps aux | grep zabbix_agentd
```

Просмотр, установлен ли Zabbix агент из пакетов:

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

Просмотр списка файлов, которые установлены пакетом установщика (обратите внимание, начальный символ «/» не отображается в выводе ниже):

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

Останов Zabbix агента, если он запущен при помощи `launchctl`:

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

Удаление файлов (включая конфигурацию и журналы), которые были установлены пакетом установщика:

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

Забыть, что Zabbix агент был установлен:

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

Сборка агента Zabbix на macOS

Обзор

Этот раздел демонстрирует, как можно собрать бинарные файлы Zabbix агента для macOS из исходного кода с/без поддержки TLS.

Предварительные требования

Вам потребуются средства разработки для командной строки (Xcode не требуется), Automake, pkg-config, а также PCRE (v8.x) или PCRE2 (v10.x). Если Вы хотите собрать бинарные файлы агента с поддержкой TLS, вам также понадобится OpenSSL или GnuTLS.

Чтобы установить Automake и pkg-config, вам потребуется менеджер пакетов Homebrew с https://brew.sh/. Чтобы его установить, откройте терминал и выполните следующую команду:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Затем установите Automake и pkg-config:

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

Подготовка библиотек PCRE, OpenSSL и GnuTLS зависит от того, каким образом будет выполняться линковка к агенту.

Если вы намерены запускать бинарные файлы агента на машине macOS, на которой уже имеются данные библиотеки, то можно использовать предварительно скомпилированные библиотеки, поставляемые Homebrew. Обычно это macOS машины, которые используют Homebrew для сборки бинарных файлов Zabbix агента или для других целей.

Если бинарные файлы агента будут использоваться на macOS машинах, на которых отсутствуют разделяемые версии библиотек, то вам потребуется скомпилировать статические версии этих библиотек из исходного кода и выполнить линковку Zabbix агента к ним.

Сборка бинарных файлов агента с разделяемыми библиотеками

Установите PCRE2 (замените *pcre2* на *pcre* в команде ниже по необходимости):

```
brew install pcre2
```

При сборке с поддержкой TLS, установите OpenSSL и / или GnuTLS:

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

Загрузите исходный код Zabbix:

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

Сборка агента без поддержки TLS:

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

Сборка агента с OpenSSL:

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

Сборка агента с GnuTLS:

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

Сборка бинарных файлов агента со статическими библиотеками без TLS

Предположим, что статические библиотеки PCRE уже установлены в `$HOME/static-libs`. Мы будем использовать PCRE2 10.39.

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

Загрузите и соберите PCRE с поддержкой свойств Юникод:

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

Загрузите исходный код Zabbix и соберите агента:

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

Сборка бинарных файлов агента со статическими библиотеками с OpenSSL

После успешной сборки OpenSSL рекомендуется выполнить `make test`. Иногда даже после успешной сборки тесты завершаются с ошибками. Если тесты завершаются с ошибками, проблемы необходимо рассмотреть и исправить перед тем как продолжить.

Предположим, что статические библиотеки PCRE и OpenSSL будут установлены в `$HOME/static-libs`. Мы будем использовать PCRE2 10.39 и OpenSSL 1.1.1a.

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

Давайте соберем статические библиотеки в `static-libs-source`:

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

Загрузите и соберите PCRE с поддержкой свойств Юникод:

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

Загрузите и соберите OpenSSL:

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

Загрузите исходный код Zabbix и соберите агента:

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

Сборка бинарных файлов агента со статическими библиотеками с GnuTLS

GnuTLS зависит от низкоуровневой криптографической библиотеки Nettle и арифметической библиотеки GMP. Вместо использования полновесной библиотеки GMP, в этом руководстве будет использоваться mini-gmp, который включён в Nettle.

После успешной сборки GnuTLS и Nettle рекомендуется выполнить `make test`. Иногда даже после успешной сборки тесты завершаются с ошибками. Если тесты завершаются с ошибками, проблемы необходимо рассмотреть и исправить перед тем как продолжить.

Предположим, что статические библиотеки PCRE, Nettle и GnuTLS будут установлены в `$HOME/static-libs`. Мы будем использовать PCRE2 10.39, Nettle 3.4.1 и GnuTLS 3.6.5.

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

Давайте соберём статические библиотеки в `static-libs-source`:

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

Загрузите и соберите Nettle:

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

Загрузите и соберите GnuTLS:

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

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

Загрузите исходный код Zabbix и соберите агента:

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

Руководство по развертыванию сервера Zabbix в AWS

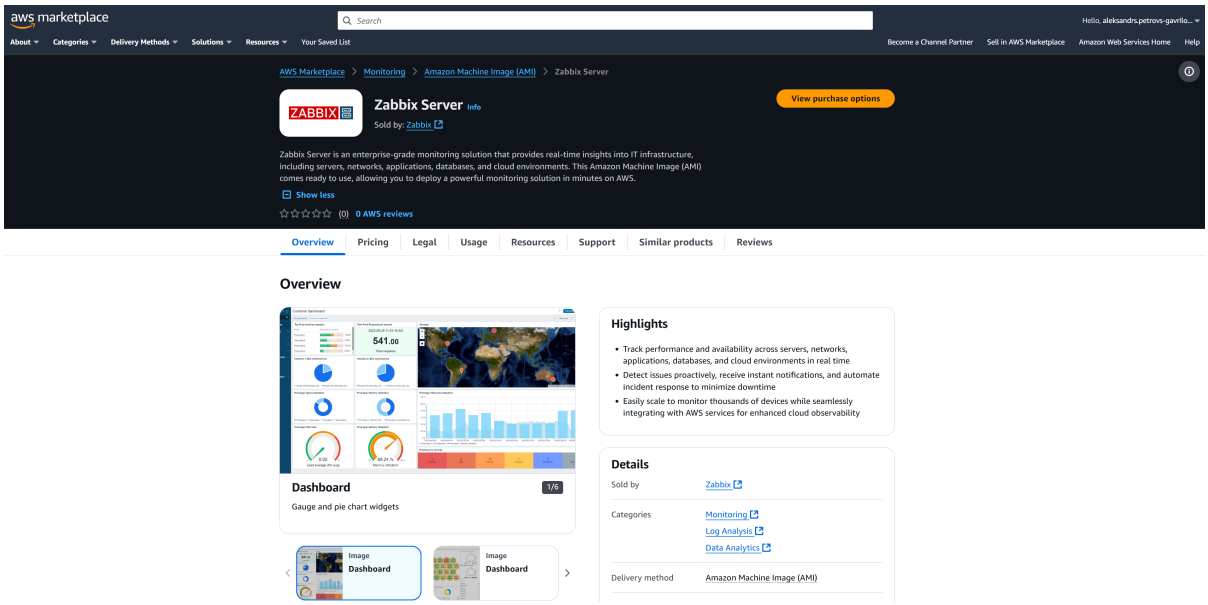
Обзор

На этой странице описывается развертывание сервера Zabbix с использованием AWS.

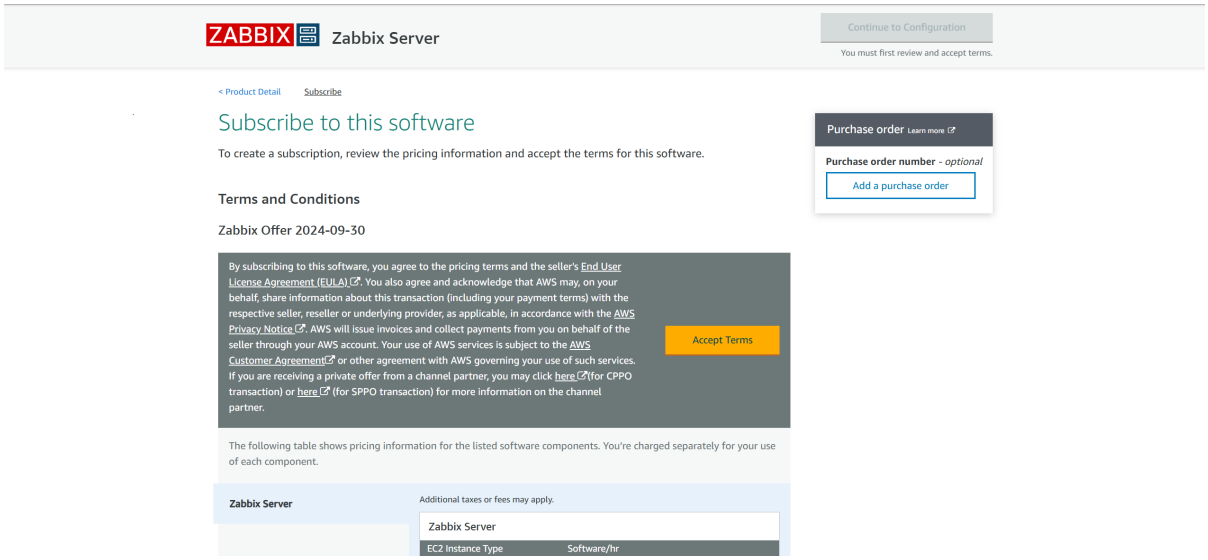
> **Attention:**
> На платформе доступны для развертывания несколько версий Zabbix. Обязательно ознакомьтесь с документацией для той же версии Zabbix, которую вы развернули. Документацию для разных версий Zabbix можно выбрать в раскрывающемся меню версий.

Развертывание устройства

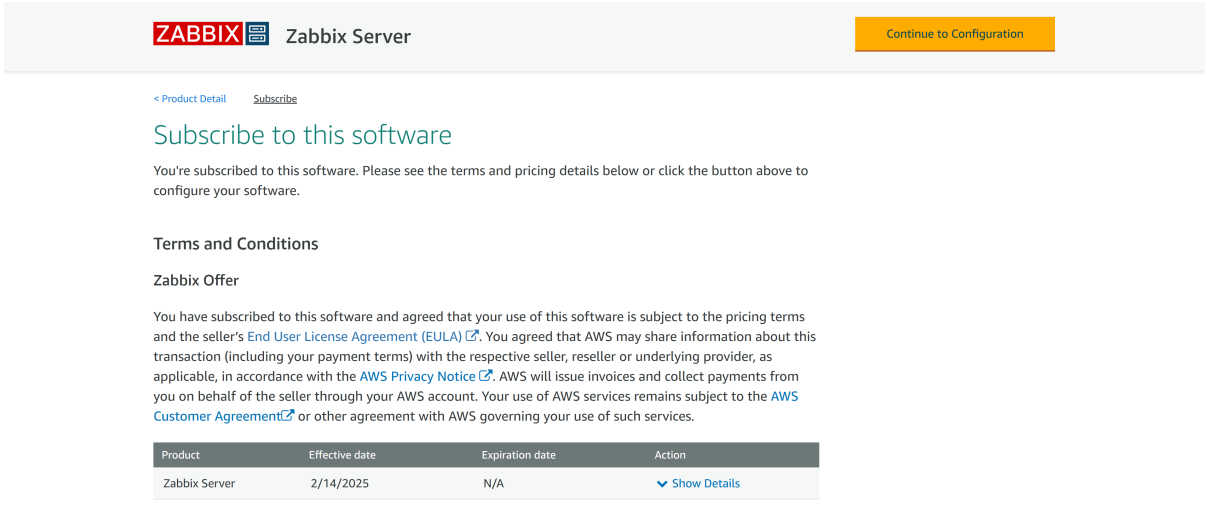
Чтобы развернуть устройство Zabbix, сначала перейдите на страницу Zabbix в AWS Marketplace. Затем нажмите *View purchase options*, чтобы развернуть образ и перейти на страницу подписки.



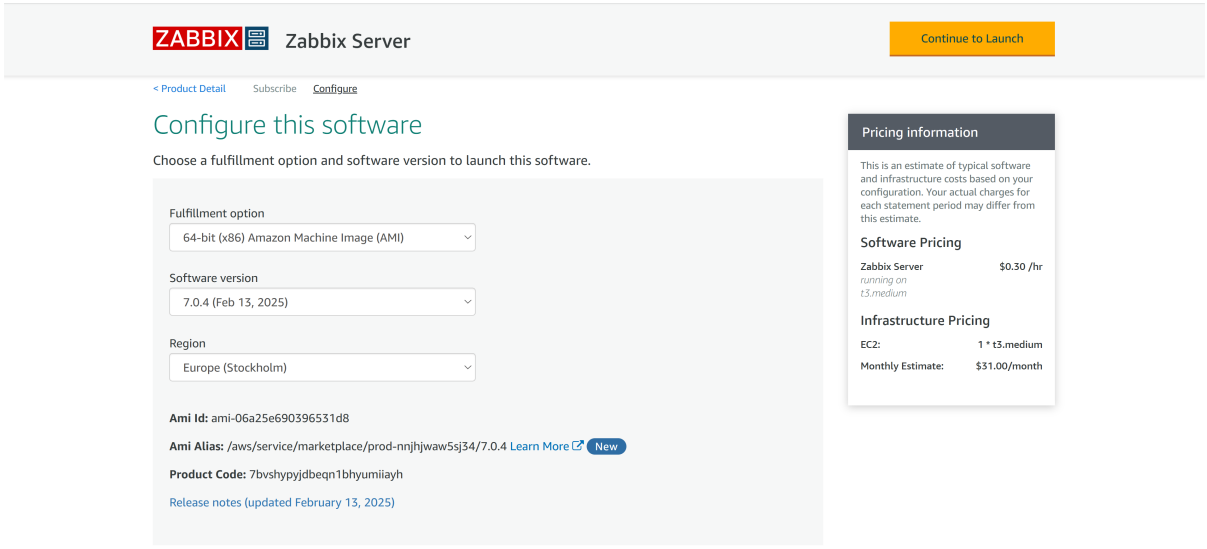
Нажмите кнопку *Accept terms*, чтобы принять *Terms and Conditions*.



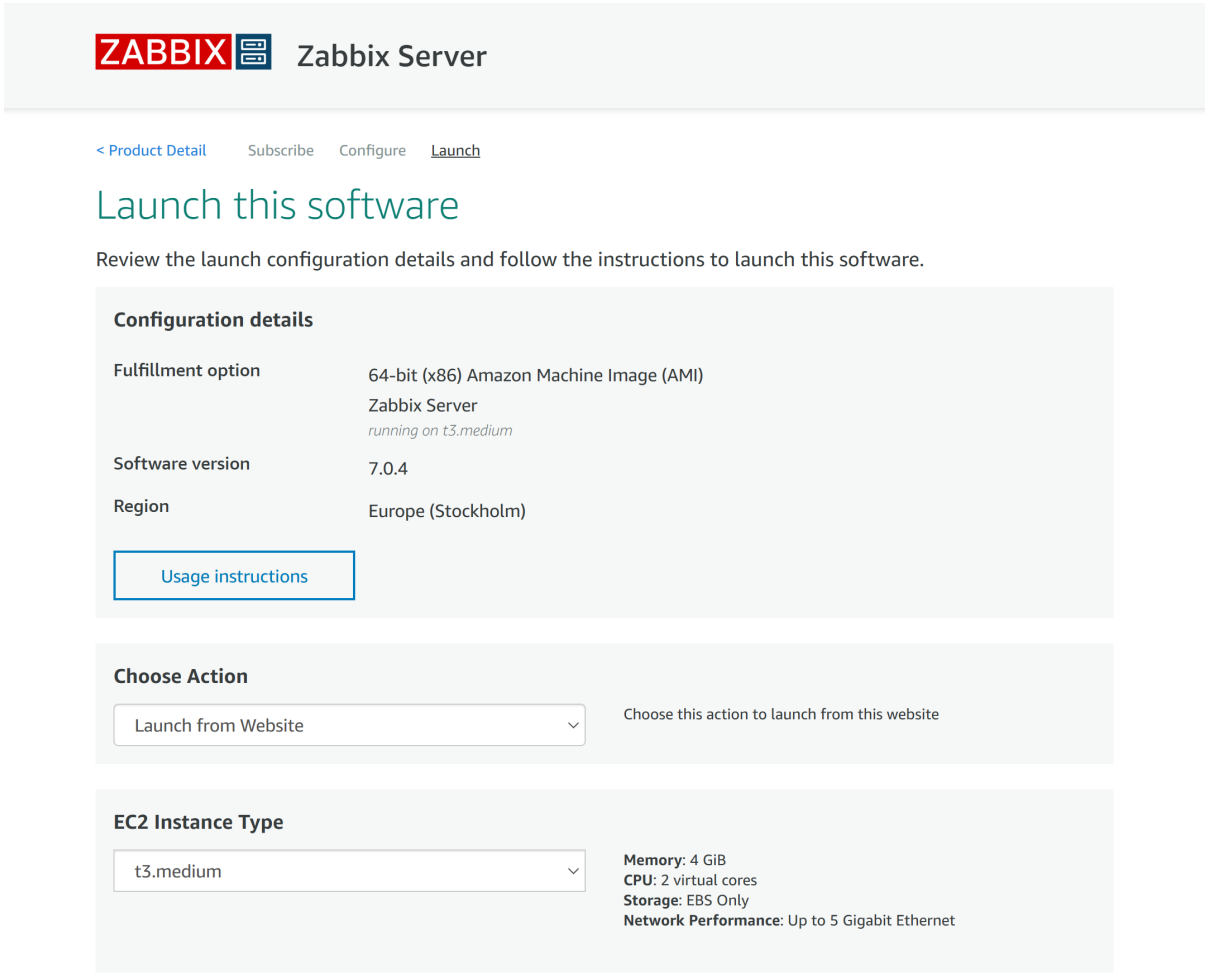
Нажмите кнопку *Continue to Configuration*, чтобы перейти к разделу Configuration для устройства Zabbix.



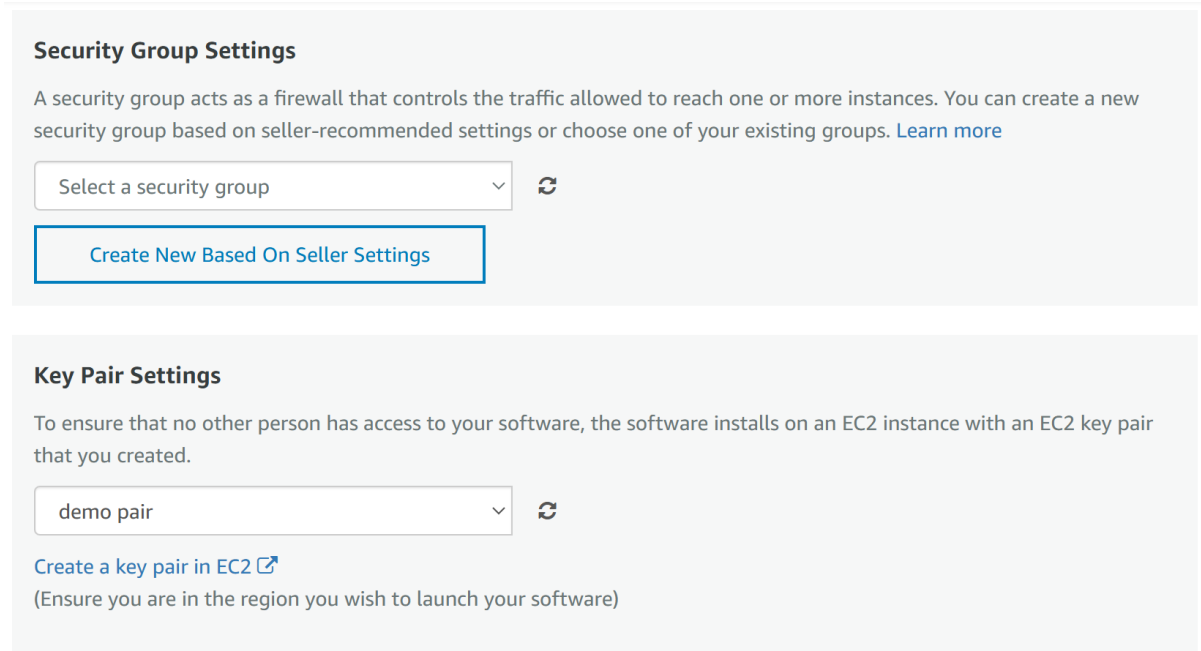
Выберите параметр Fulfilment, версию Zabbix и предпочтительный регион.



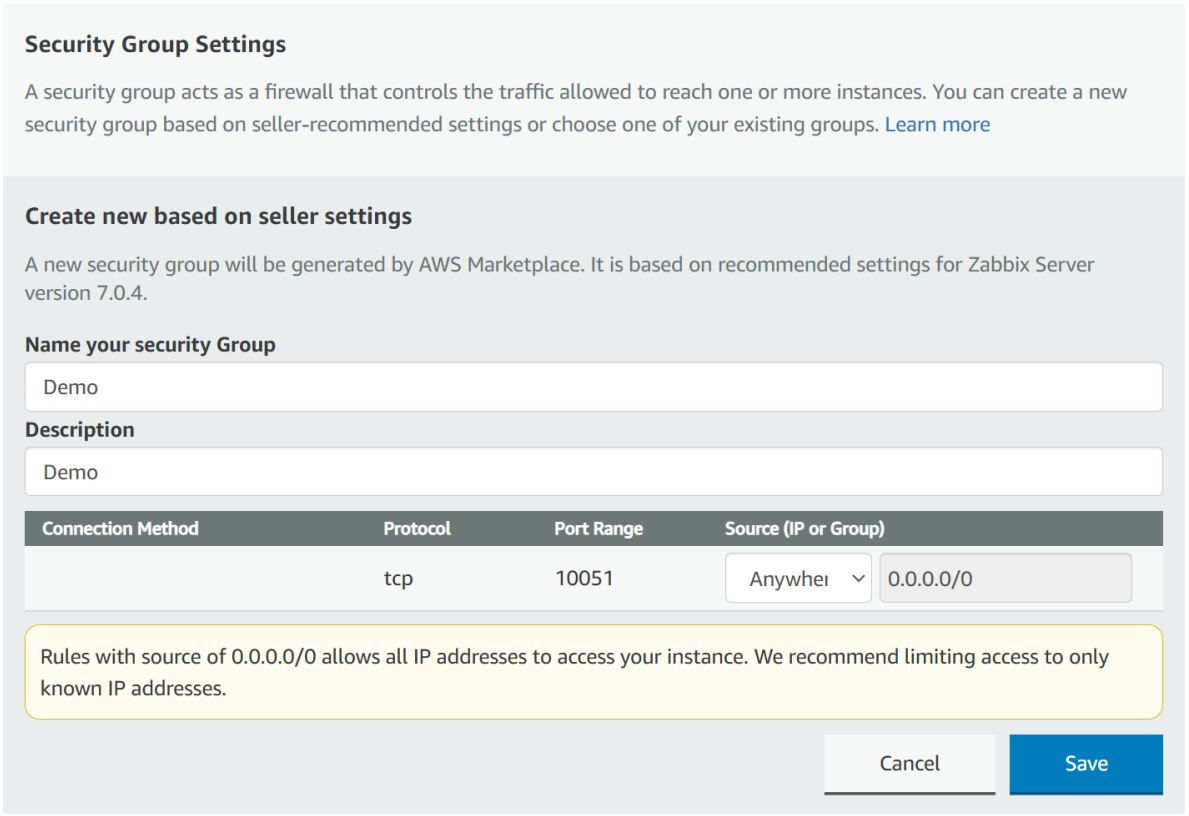
Нажмите кнопку *Continue to Launch*. На странице Launch выберите предпочтительные параметры Launch, тип экземпляра EC2, VPC и подсети.



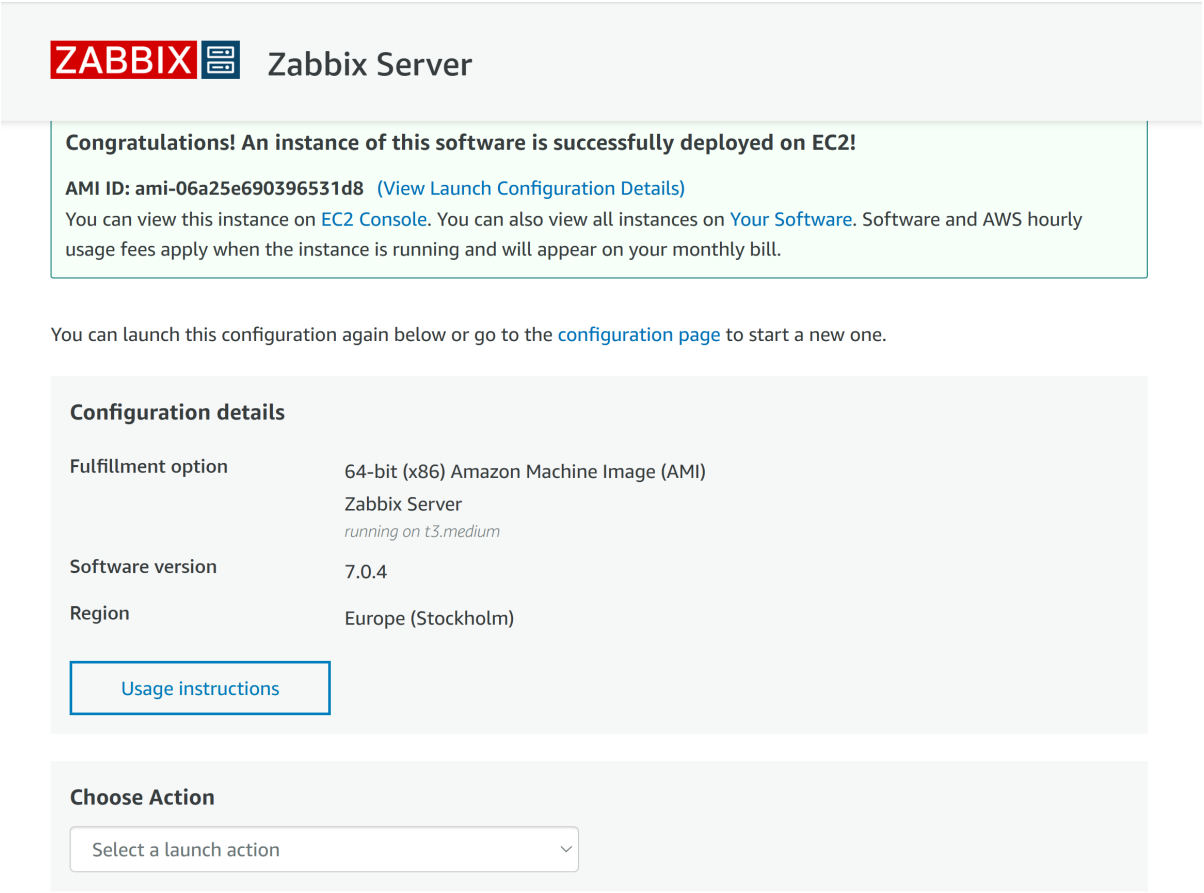
В нижней части страницы Launch укажите параметры группы безопасности и выберите или создайте пару ключей EC2 - обязательно сохраните закрытый ключ в безопасном месте!



Если выбрать *Create a new Security Group based on seller settings*, будет разрешен TCP-порт 10051 (сервер Zabbix). Примечание: если вы хотите получить доступ к экземпляру через SSH и HTTP/HTTPS (веб-интерфейс), необходимо также разрешить подключения SSH и HTTP/HTTPS в параметрах группы безопасности.

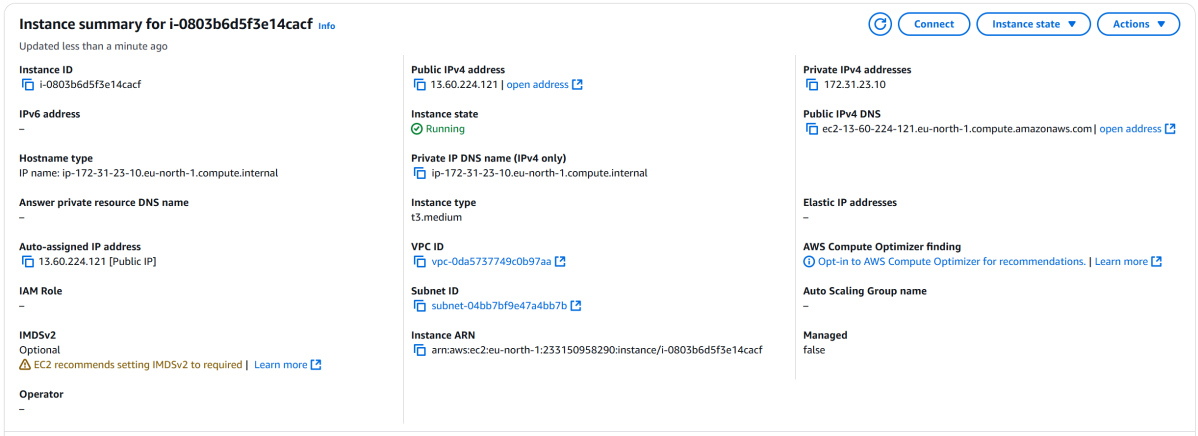


Поздравляем, устройство Zabbix успешно развернуто.

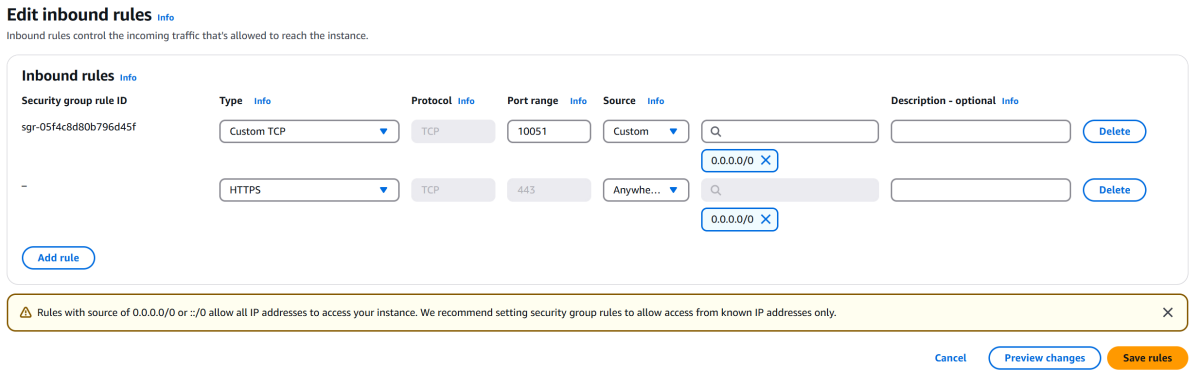


Доступ к устройству

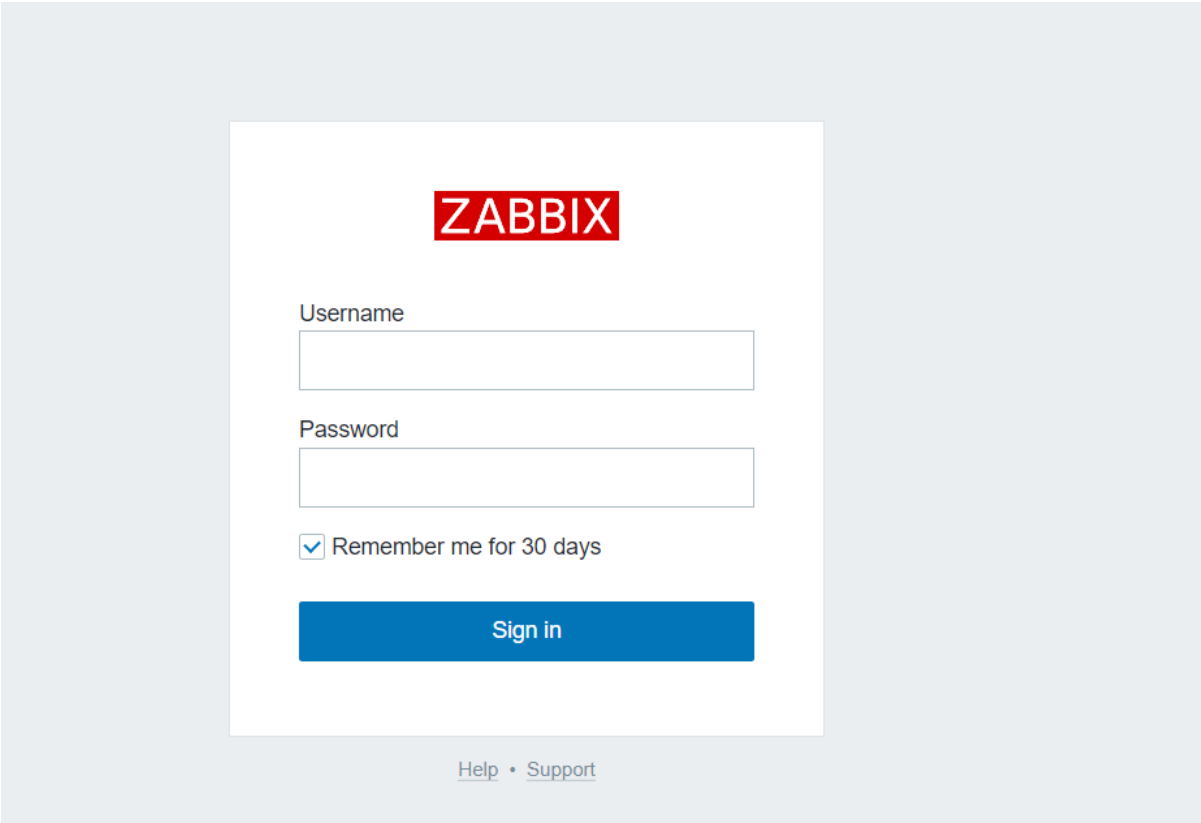
Откройте раздел *Instances* и выберите недавно развернутый экземпляр устройства Zabbix.



Чтобы получить доступ к веб-интерфейсу Zabbix с использованием общедоступного IPv4-адреса или общедоступного DNS-имени IPv4 экземпляра, разрешите входящий HTTPS-трафик через порт 443 в группе безопасности.



Теперь можно получить доступ к веб-интерфейсу Zabbix, открыв общедоступный IPv4-адрес или общедоступное DNS-имя IPv4 устройства Zabbix.



Чтобы получить пароль, необходимо подключиться к машине через последовательную консоль или SSH. Имя пользователя SSH по умолчанию — rocky. Не забудьте использовать пару ключей, выбранную или созданную ранее. Обратите внимание, что создание учетных данных может занять некоторое время. Создание учетных данных может занять несколько минут. Подождите несколько минут, прежде чем пытаться подключиться.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

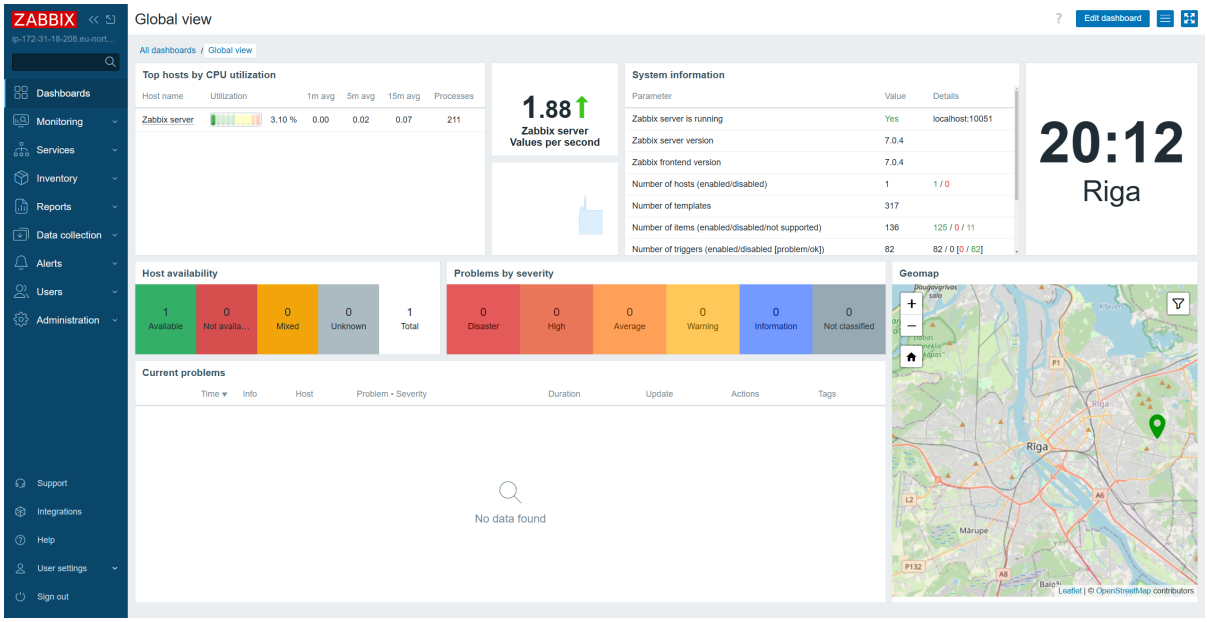
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

Теперь эту информацию можно использовать для входа в веб-интерфейс Zabbix.



Доступ к базе данных

Вы можете получить доступ к базе данных Zabbix с помощью пользователя `root` без ввода пароля.

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Учетные данные хранятся в файле конфигурации `/root/.my.cnf`.

Часовой пояс веб-интерфейса Zabbix по умолчанию можно изменить в разделе *Администрирование > Общие параметры > Веб-интерфейс*.

Руководство по развертыванию прокси Zabbix в AWS

Обзор

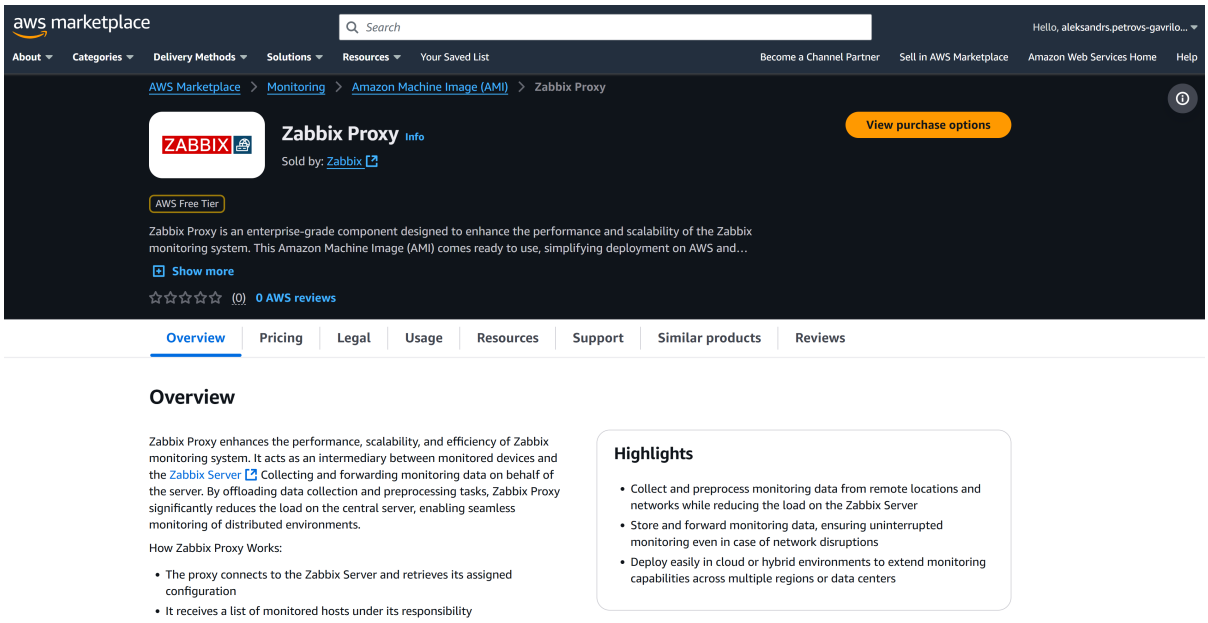
На этой странице описывается развертывание прокси Zabbix с использованием AWS.

Attention:
Для развертывания на платформе доступны несколько версий Zabbix. Обязательно ознакомьтесь с документацией для той же версии Zabbix, которую вы развернули. Документацию для разных версий Zabbix можно выбрать в раскрывающемся меню версий.

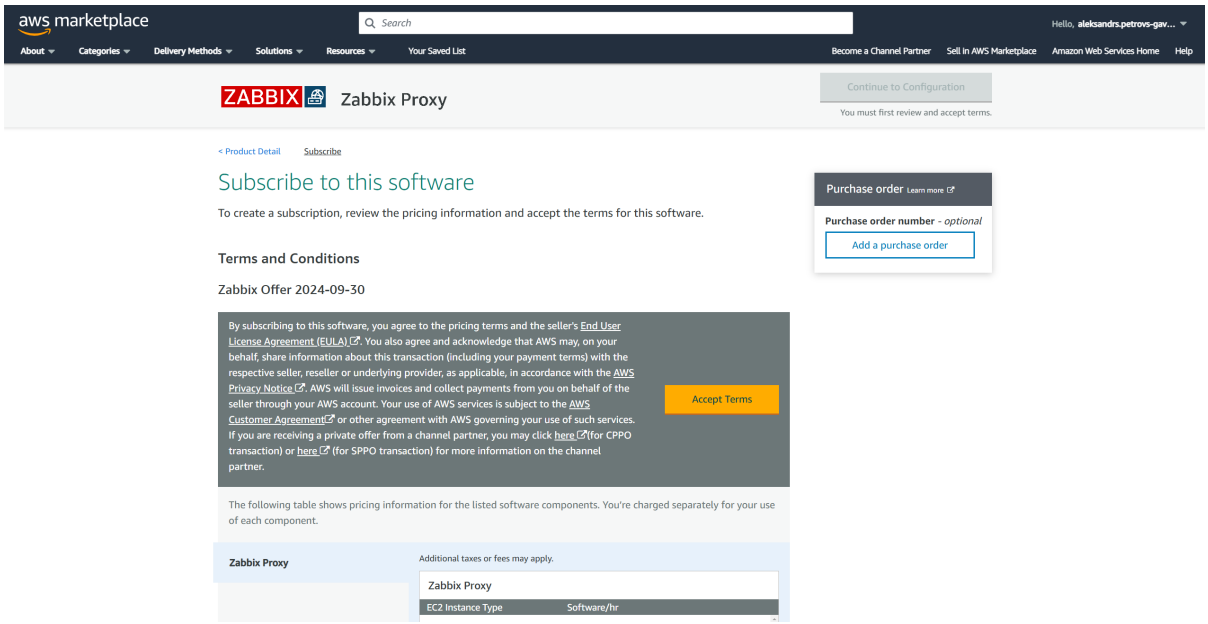
Развертывание прокси Zabbix

Attention:
Для использования компонента прокси Zabbix требуется работающий экземпляр сервера Zabbix.

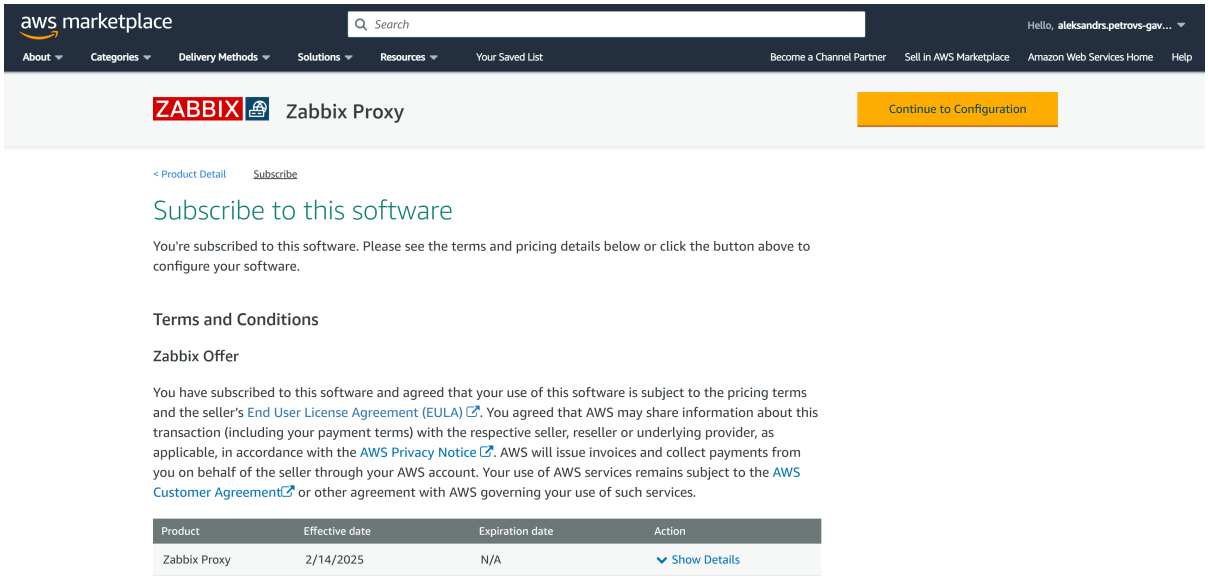
Чтобы развернуть прокси Zabbix, сначала перейдите на страницу Zabbix в AWS Marketplace. Затем нажмите *View purchase options*, чтобы развернуть образ и перейти на страницу подписки.



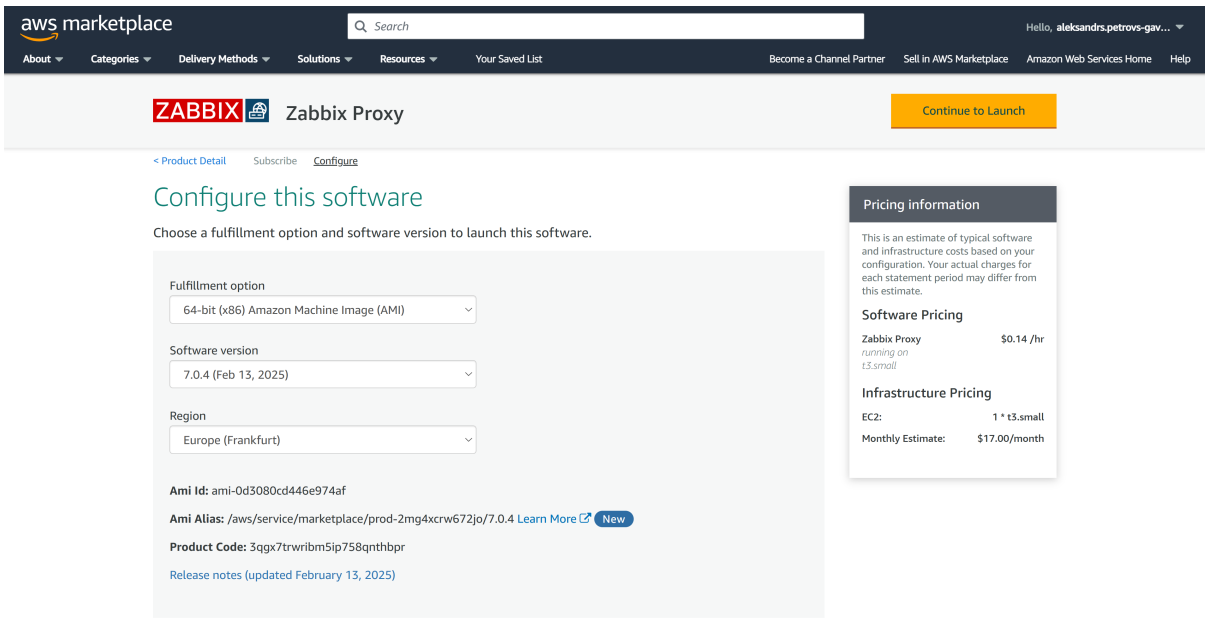
Нажмите кнопку *Accept terms*, чтобы принять *Terms and Conditions*.



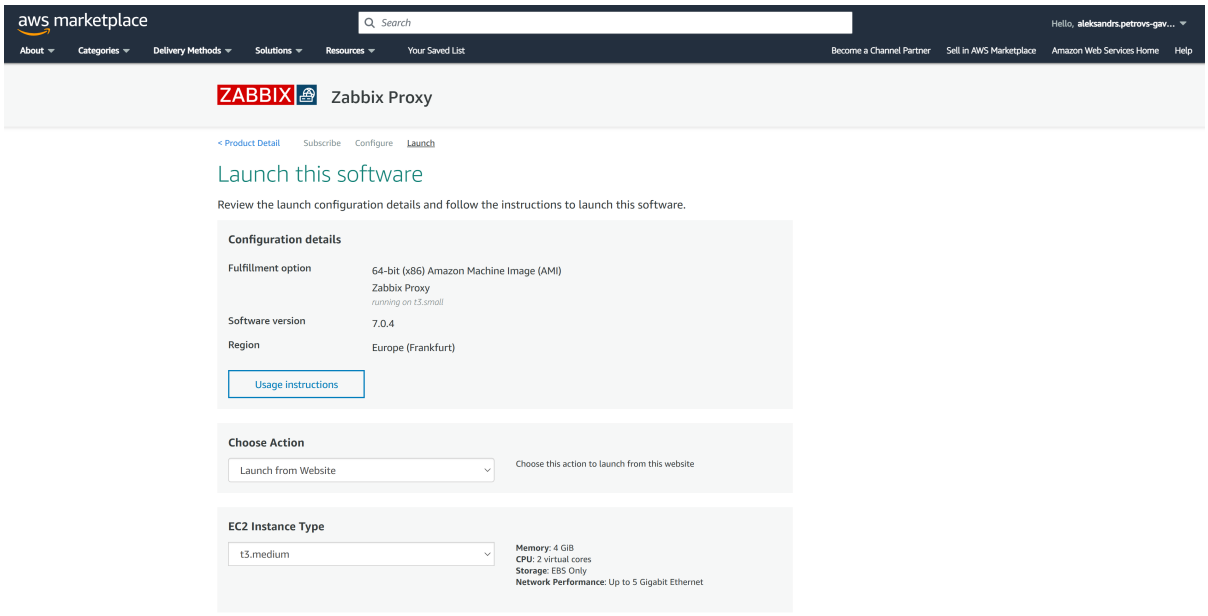
Нажмите кнопку *Continue to Configuration*, чтобы перейти к разделу Configuration для прокси Zabbix.



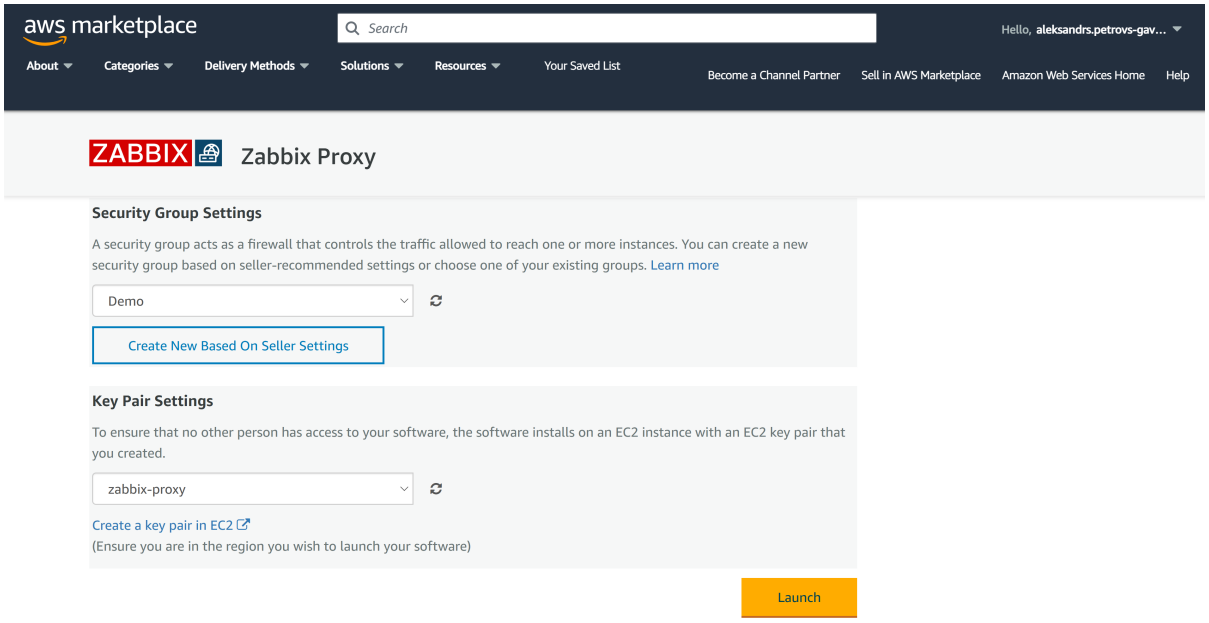
Выберите вариант Fulfilment, версию Zabbix и предпочтительный регион.



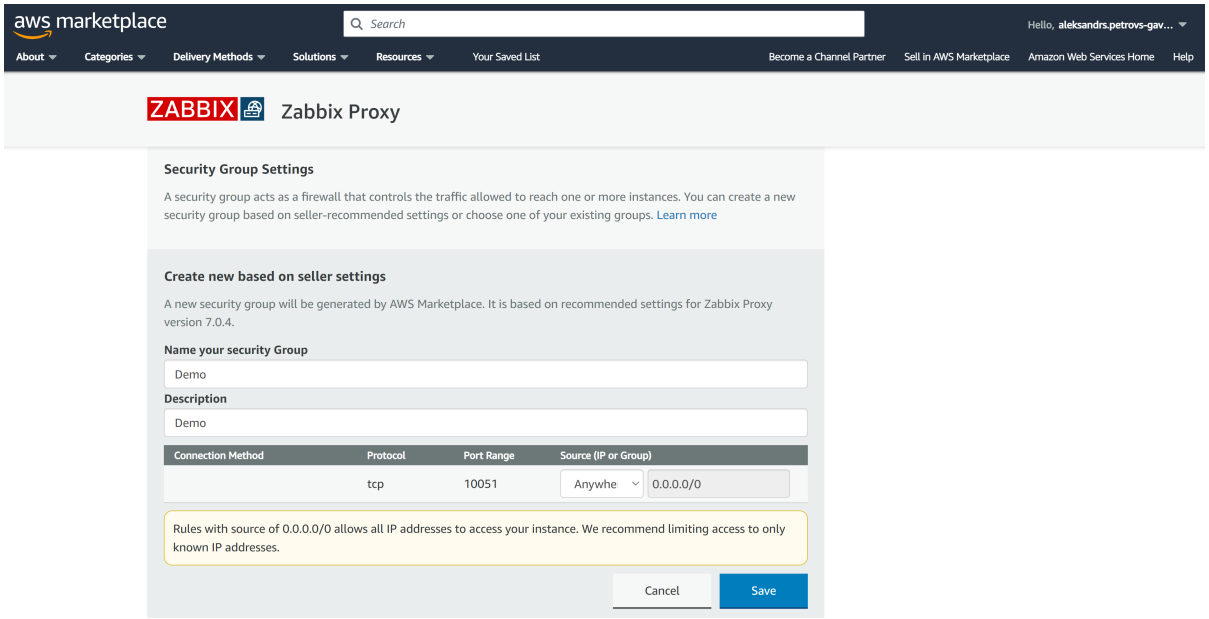
Нажмите кнопку *Continue to Launch*. На странице Launch выберите предпочтительные параметры Launch, тип экземпляра EC2, VPC и подсети.



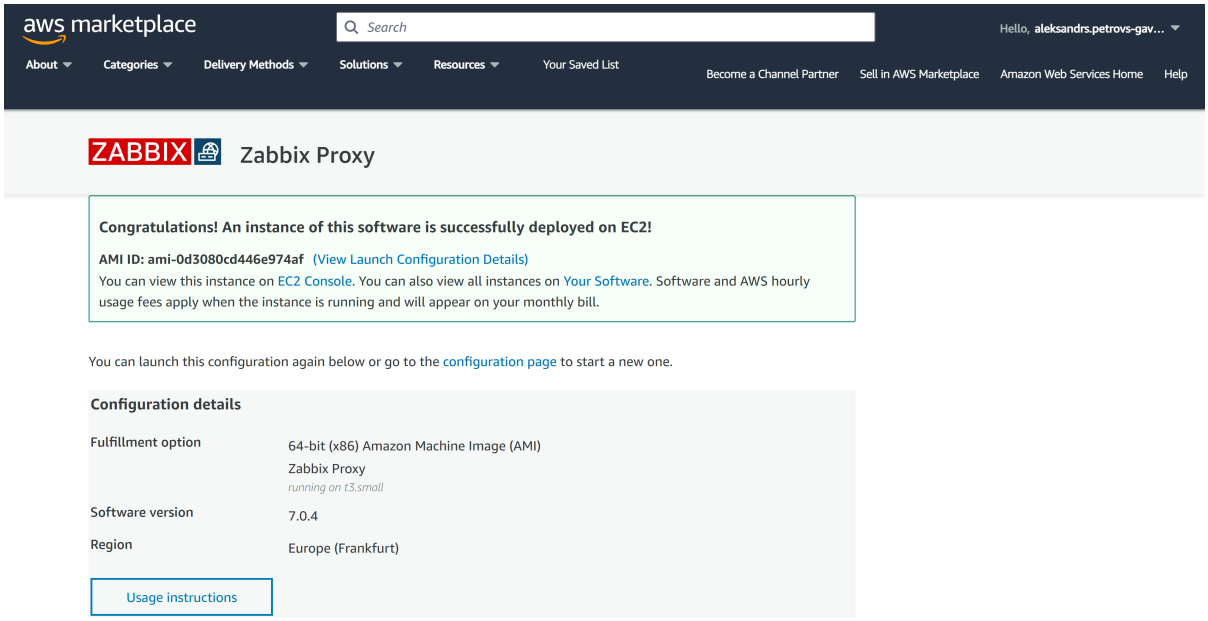
В нижней части страницы Launch укажите параметры группы безопасности и выберите или создайте пару ключей EC2. Обязательно сохраните закрытый ключ в безопасном месте.



Если выбрать *Create a new Security Group based on the Seller settings*, будет разрешен TCP-порт 10051 (сервер Zabbix). Примечание: если вы хотите получить доступ к экземпляру через SSH, необходимо также разрешить SSH-подключения в настройках группы безопасности.

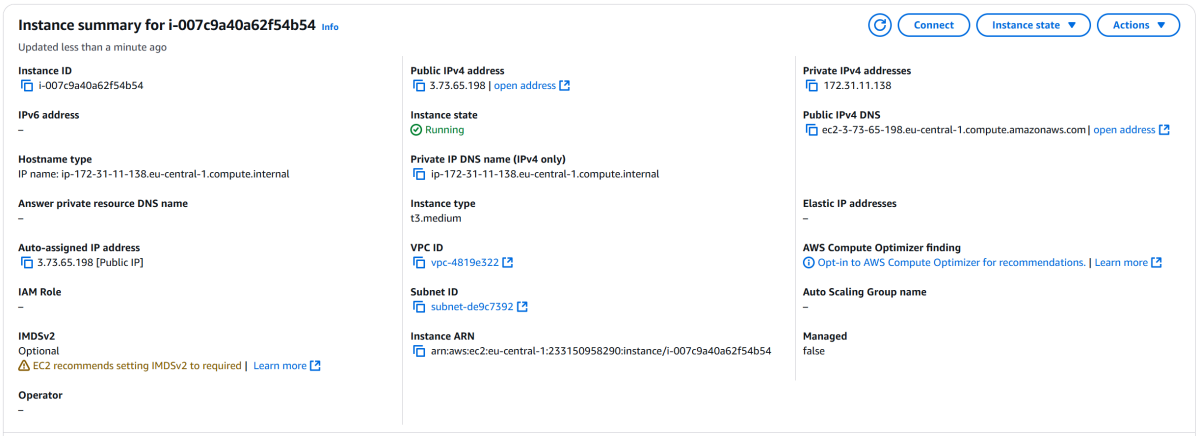


Наконец, нажмите кнопку *Launch*. Поздравляем! Развертывание прокси Zabbix началось.



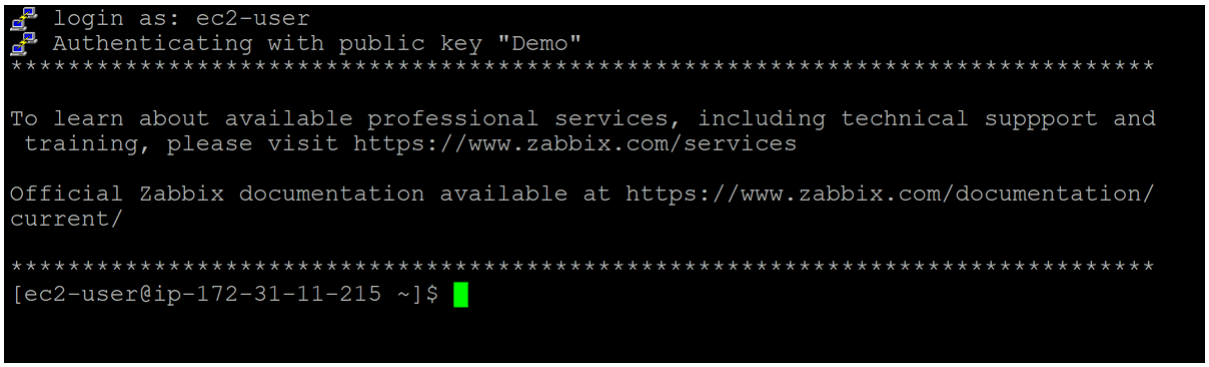
Настройка и доступ к прокси Zabbix

Откройте раздел Instances в консоли EC2 и выберите недавно развернутый экземпляр прокси Zabbix.



Подключитесь к общедоступному адресу IPv4 или общедоступному DNS-имени IPv4 прокси Zabbix по протоколу SSH. Имя пользователя SSH по умолчанию — `rocky`. Не забудьте использовать пару ключей, созданную ранее.

Генерация учетных данных может занять несколько минут. Подождите несколько минут, прежде чем пытаться подключиться.



Файл конфигурации прокси Zabbix находится в `/etc/zabbix/zabbix_proxy.conf`.

Файл конфигурации можно открыть в любом удобном текстовом редакторе, например vi:

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

Первоначальная настройка прокси заключается в изменении параметров `ProxyMode`, `Server` и `Hostname` в файле конфигурации прокси Zabbix.

Режим прокси можно изменить путем редактирования параметра `ProxyMode`. Установите `ProxyMode=0` для активного режима или `ProxyMode=1` для пассивного режима.

Параметр `Server` должен содержать IP-адрес сервера Zabbix. Например:

```
Server=192.168.50.141
```

Параметр `Hostname` должен содержать имя узла сети прокси Zabbix. Указанное здесь имя узла сети должно совпадать с именем прокси, заданным во время настройки прокси в веб-интерфейсе Zabbix. Например:

```
Hostname=EC2-Frankfurt
```

После завершения настройки перезапустите прокси Zabbix с помощью следующей команды:

```
sudo systemctl restart zabbix-proxy
```

Примечание: используйте `sudo` для редактирования файлов конфигурации и выполнения команд из командной строки.

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

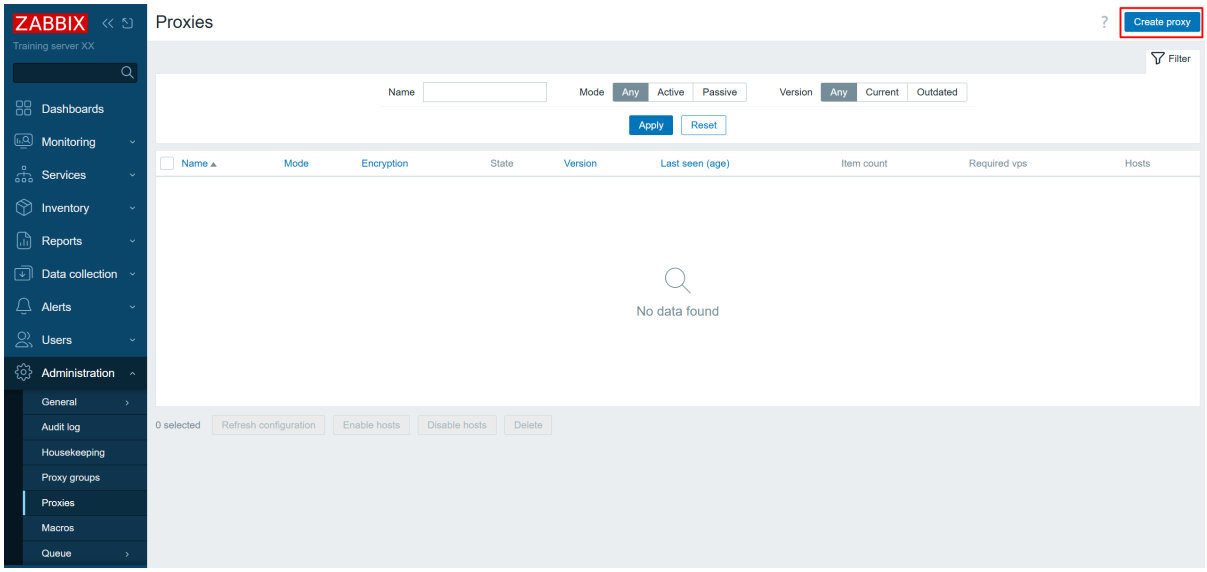
Hostname=EC2-Frankfurt
```

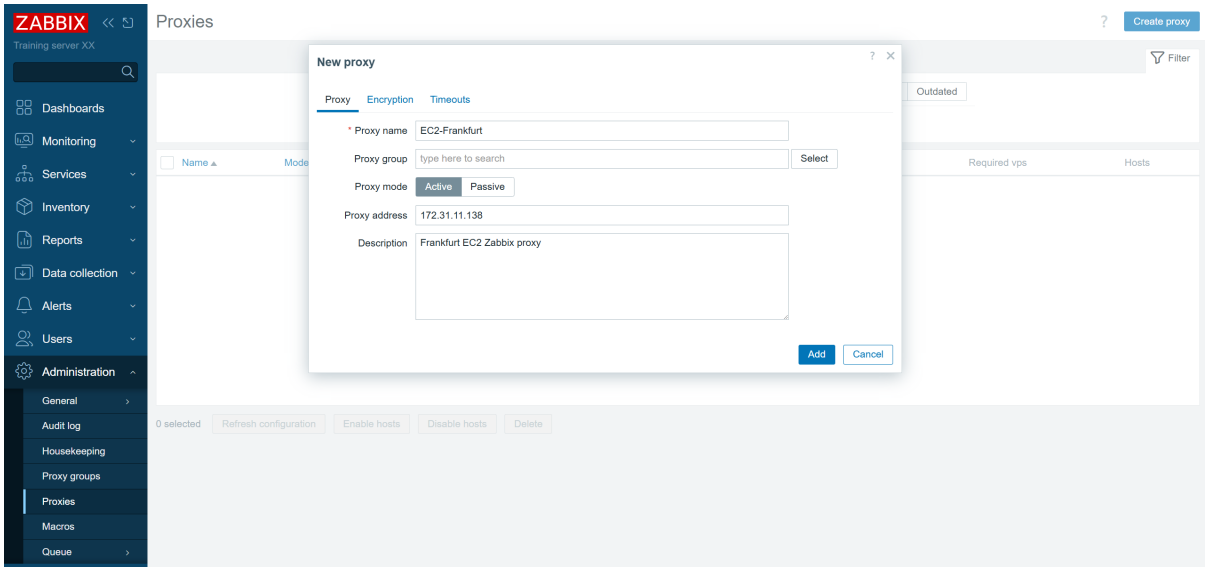
Откройте веб-интерфейс Zabbix и перейдите в раздел *Administration* > *Proxies*. Нажмите кнопку *Create proxy*.

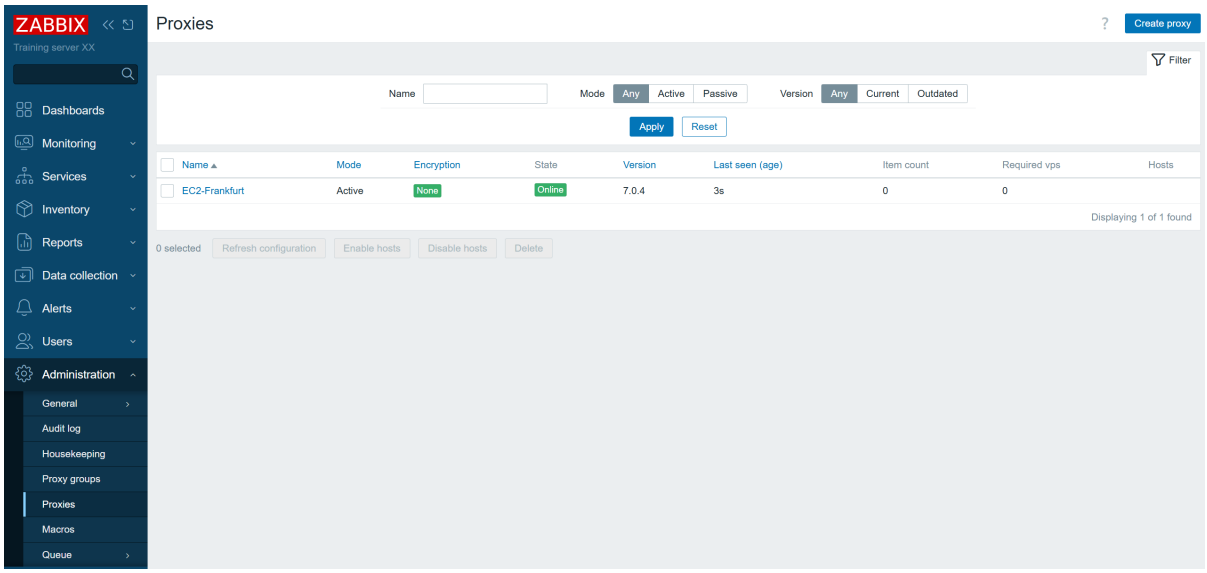
Введите то же имя прокси, что и значение параметра `Hostname` в файле конфигурации прокси.

Выберите режим прокси и укажите IP-адрес прокси (необязательно для активных прокси).

Нажмите *Create proxy*.







Расположение базы данных прокси

База данных SQLite прокси хранится в `/tmp/zabbix_proxy.sqlite3` и указывается параметром `DBName` в файле конфигурации прокси Zabbix.

Руководство по развертыванию сервера Zabbix в Azure

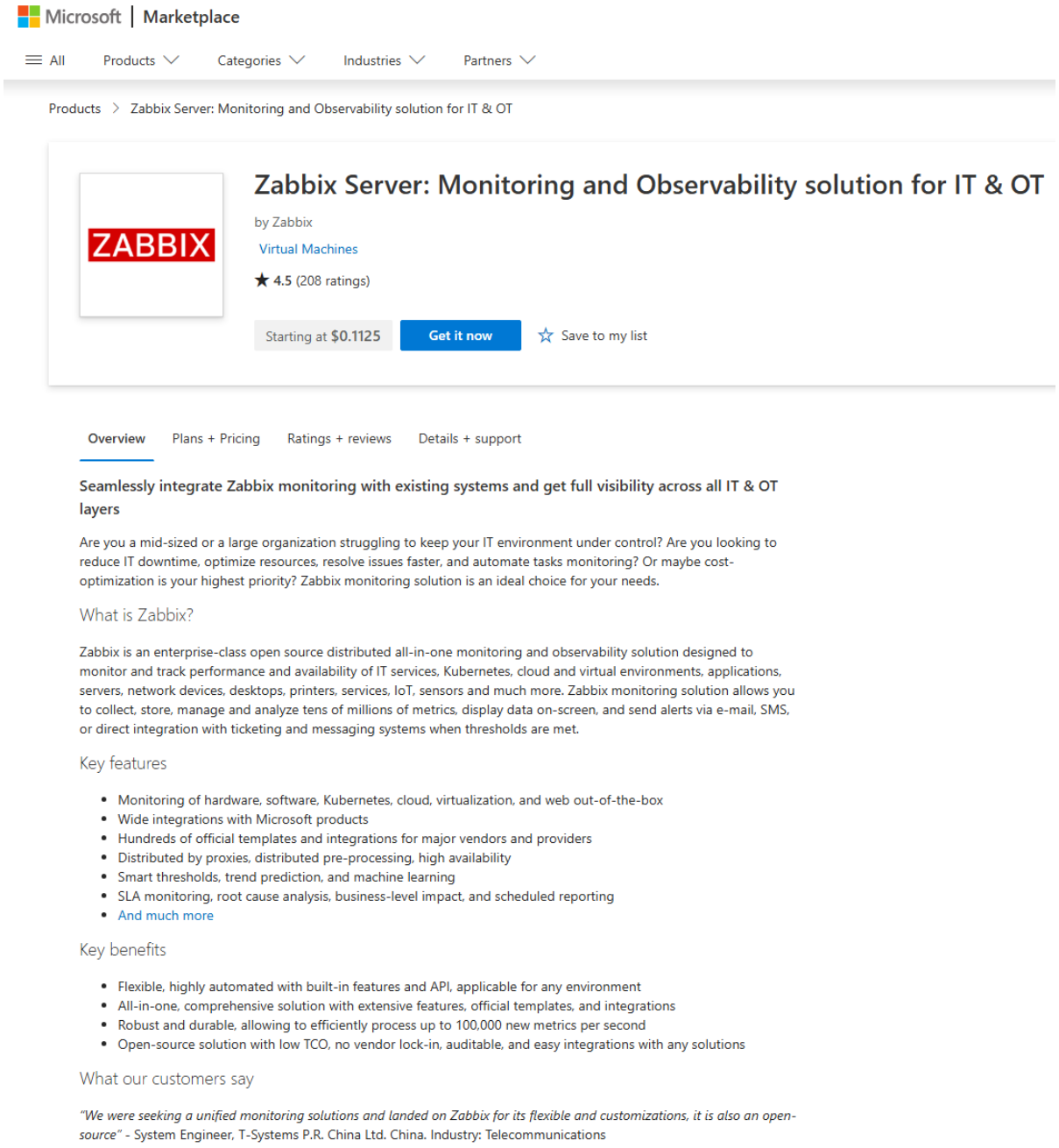
Обзор

На этой странице описывается развертывание сервера Zabbix с использованием Azure.

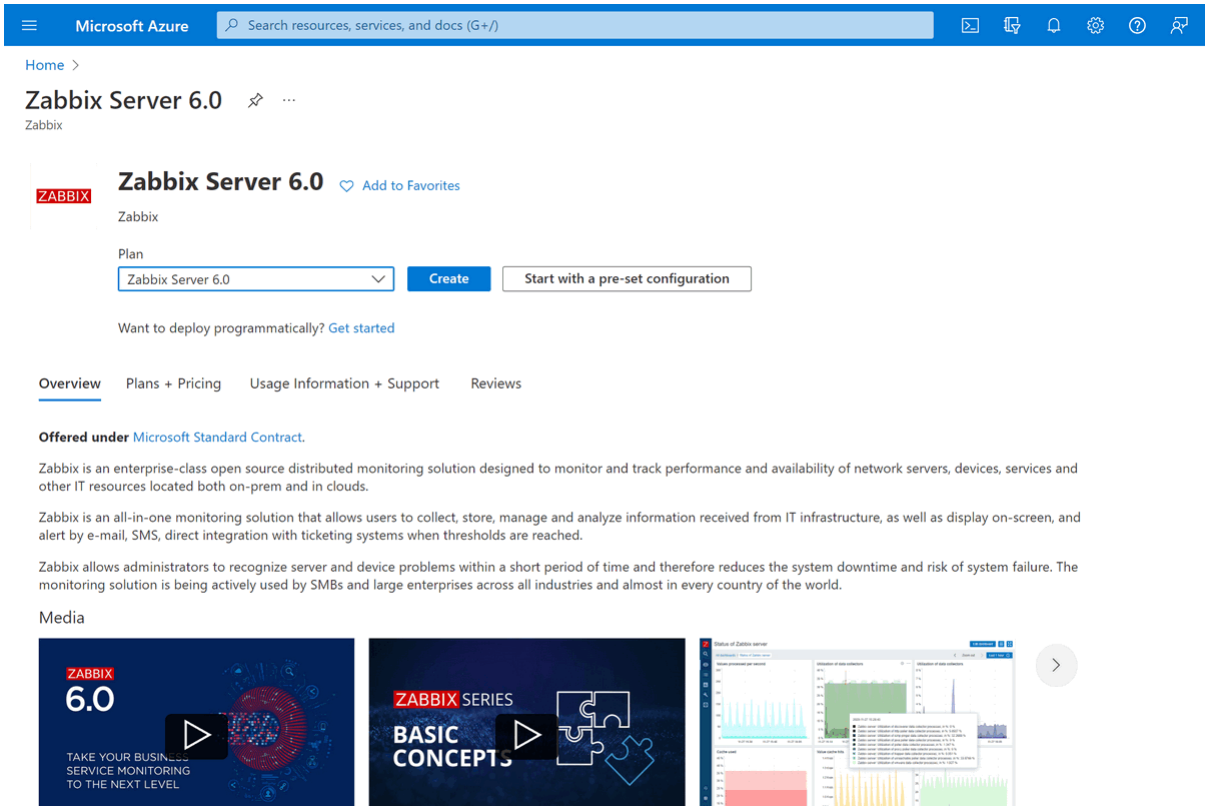
Attention:
Для развертывания на платформе доступны несколько версий Zabbix. Обязательно ознакомьтесь с документацией для той же версии Zabbix, которую вы развернули. Документацию для разных версий Zabbix можно выбрать в раскрывающемся меню версий.

Развертывание устройства

Чтобы развернуть устройство Zabbix, сначала перейдите на страницу Zabbix в Azure Marketplace. Ознакомьтесь с обзором, затем нажмите *Get it now*.



Ознакомьтесь с обзором развертывания, тарифными планами и другими сведениями. Затем нажмите *Create* или *Start with a pre-set configuration*, чтобы развернуть устройство Zabbix.

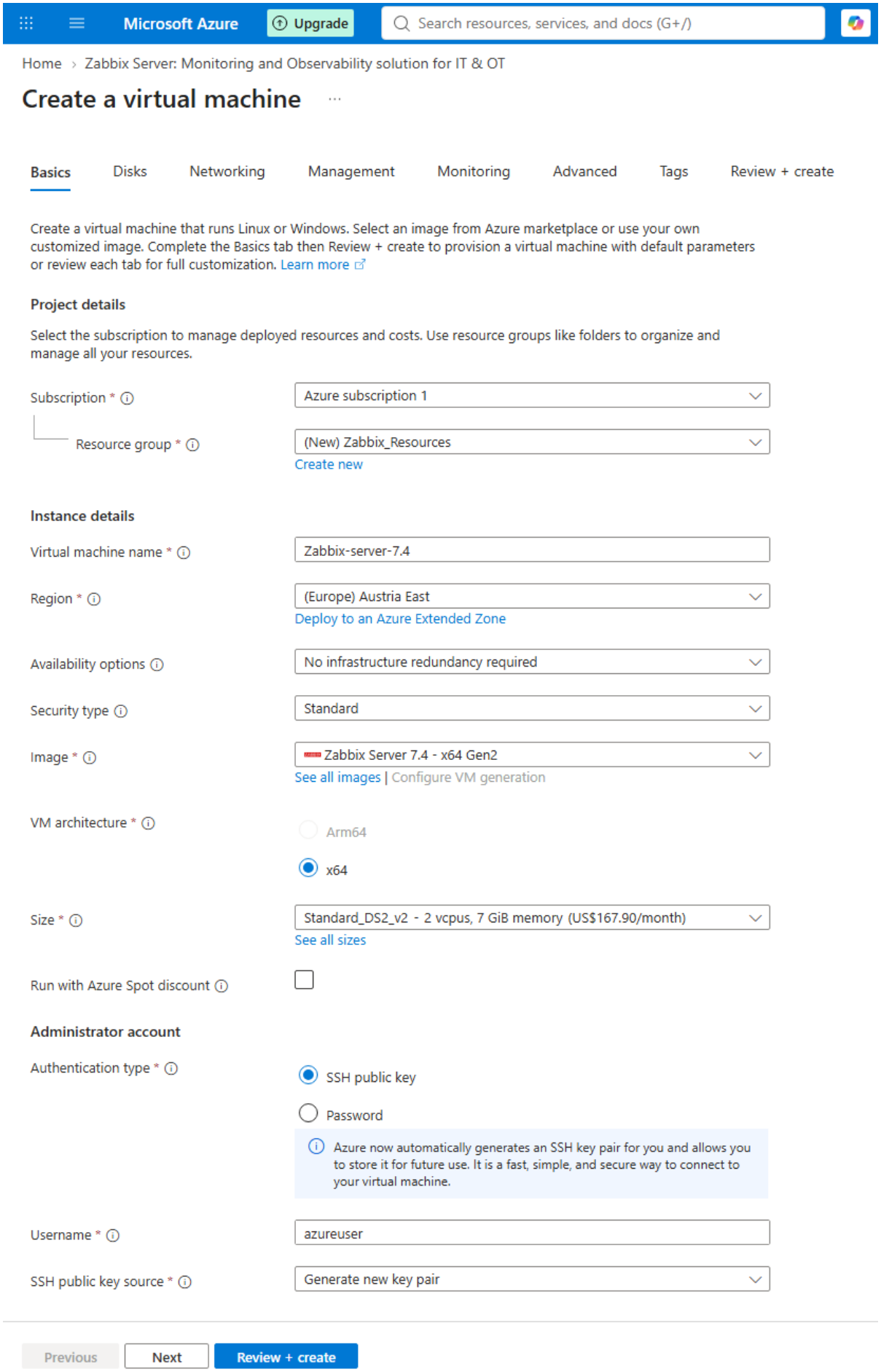


Настройте виртуальную машину, выбрав ее размер, создав учетную запись администратора и задав параметры диска, сети, управления и другие параметры. Эти параметры могут различаться в зависимости от требований вашей инфраструктуры.

В качестве примера можно использовать следующую конфигурацию:

Вкладка *Basics*:

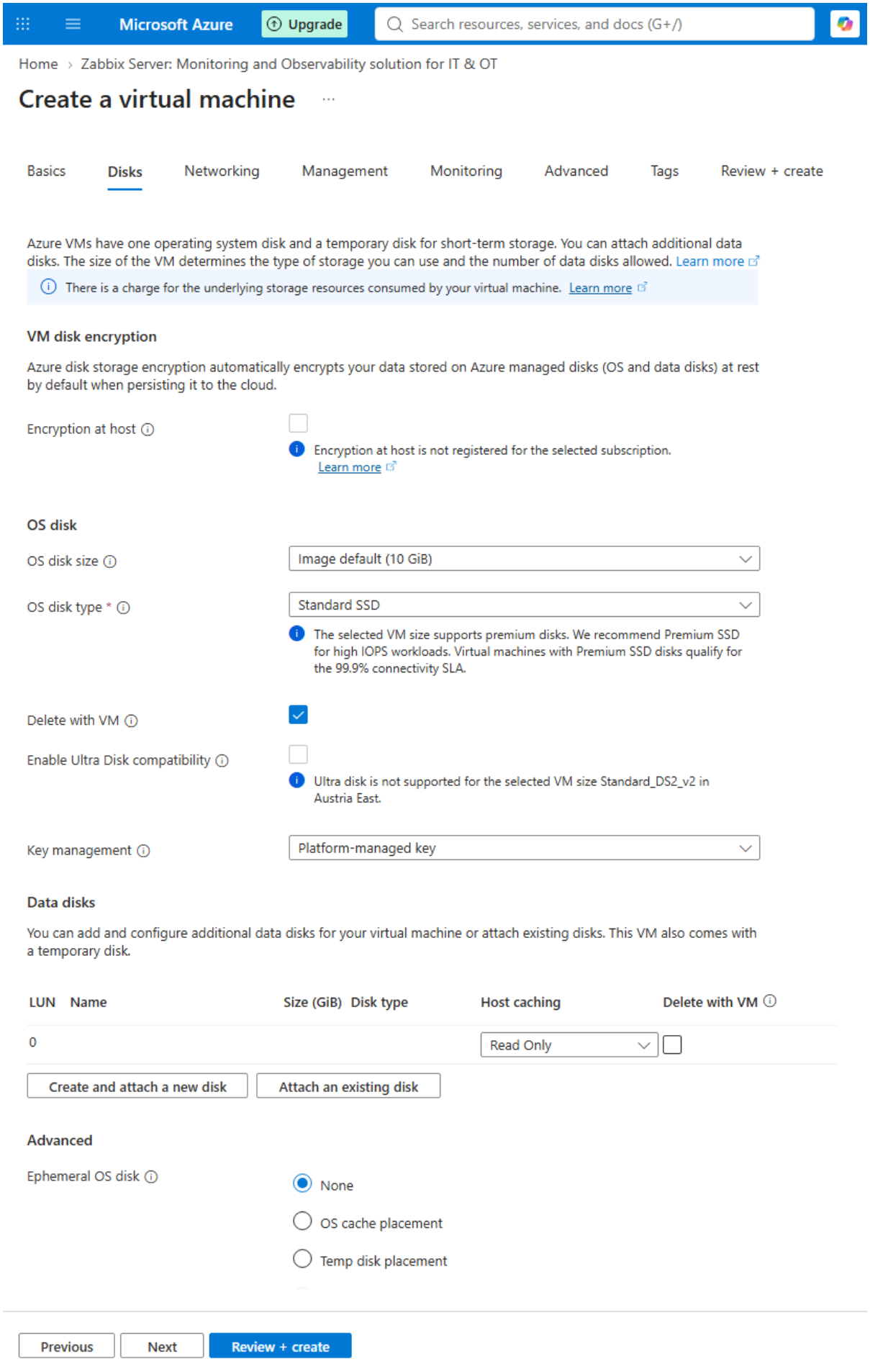
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Вкладка *Disks*:

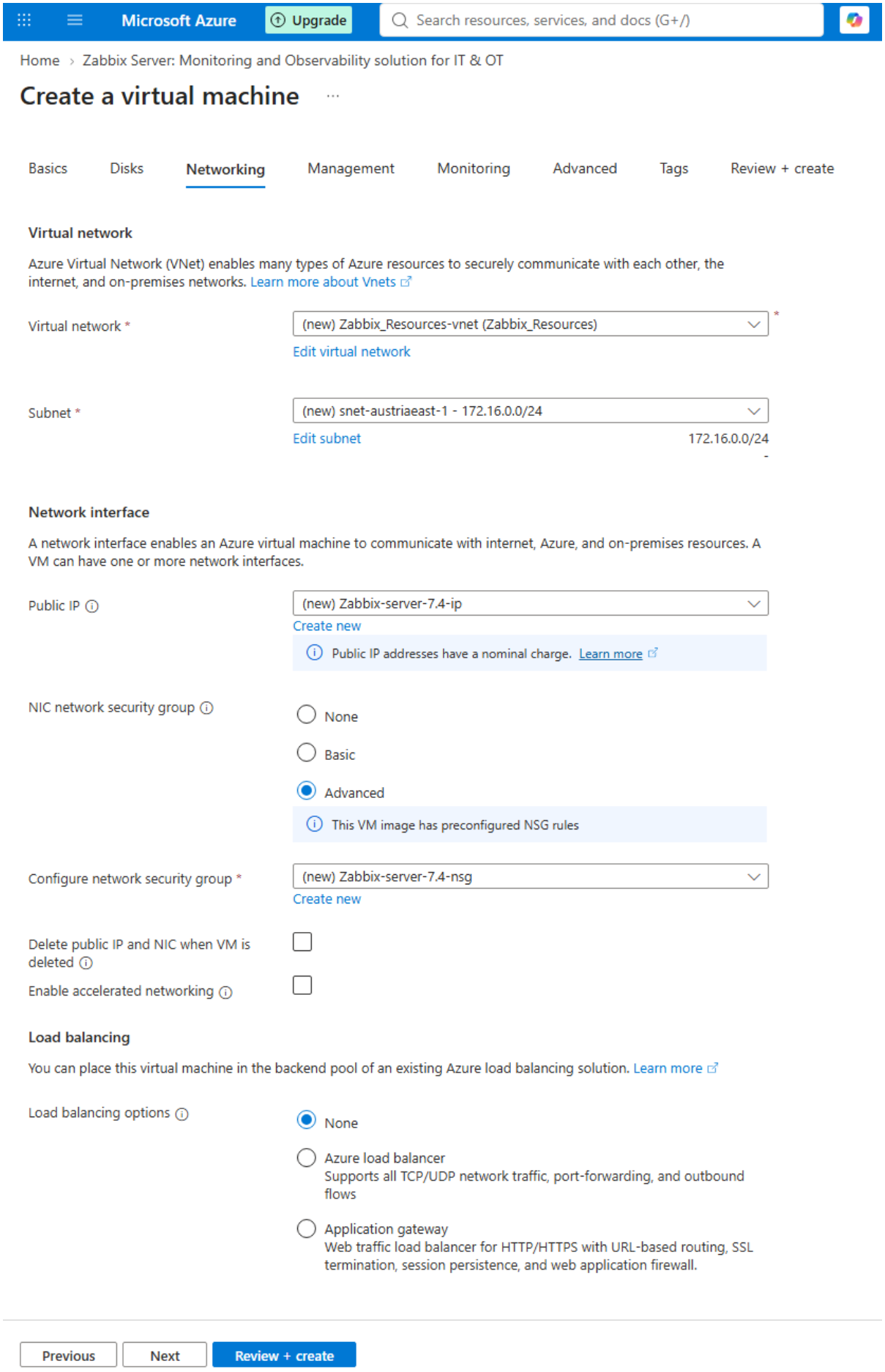
- *Encryption at host*: unchecked
- *OS disk type*: Standard SSD

- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

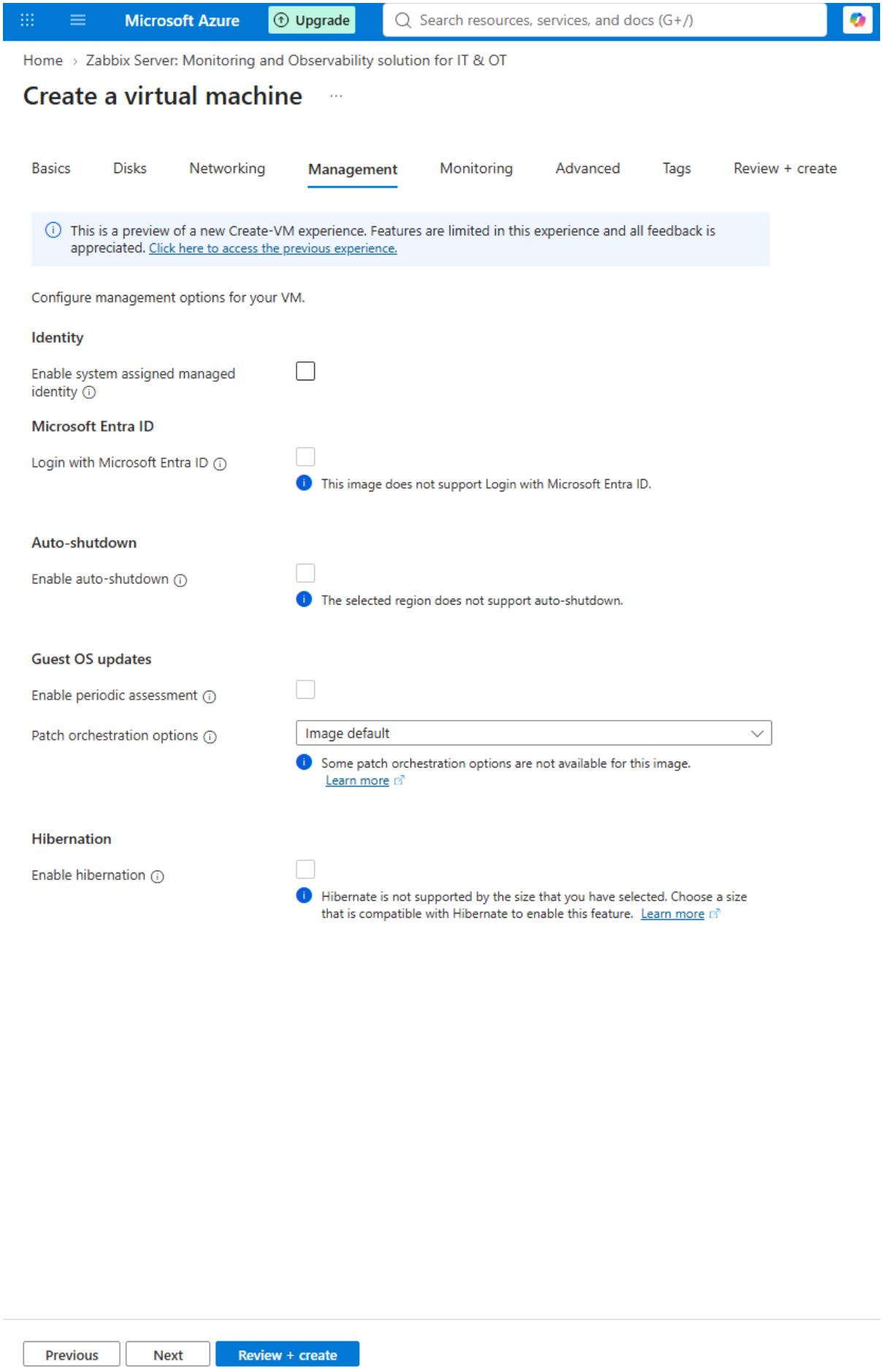


Вкладка *Networking*:

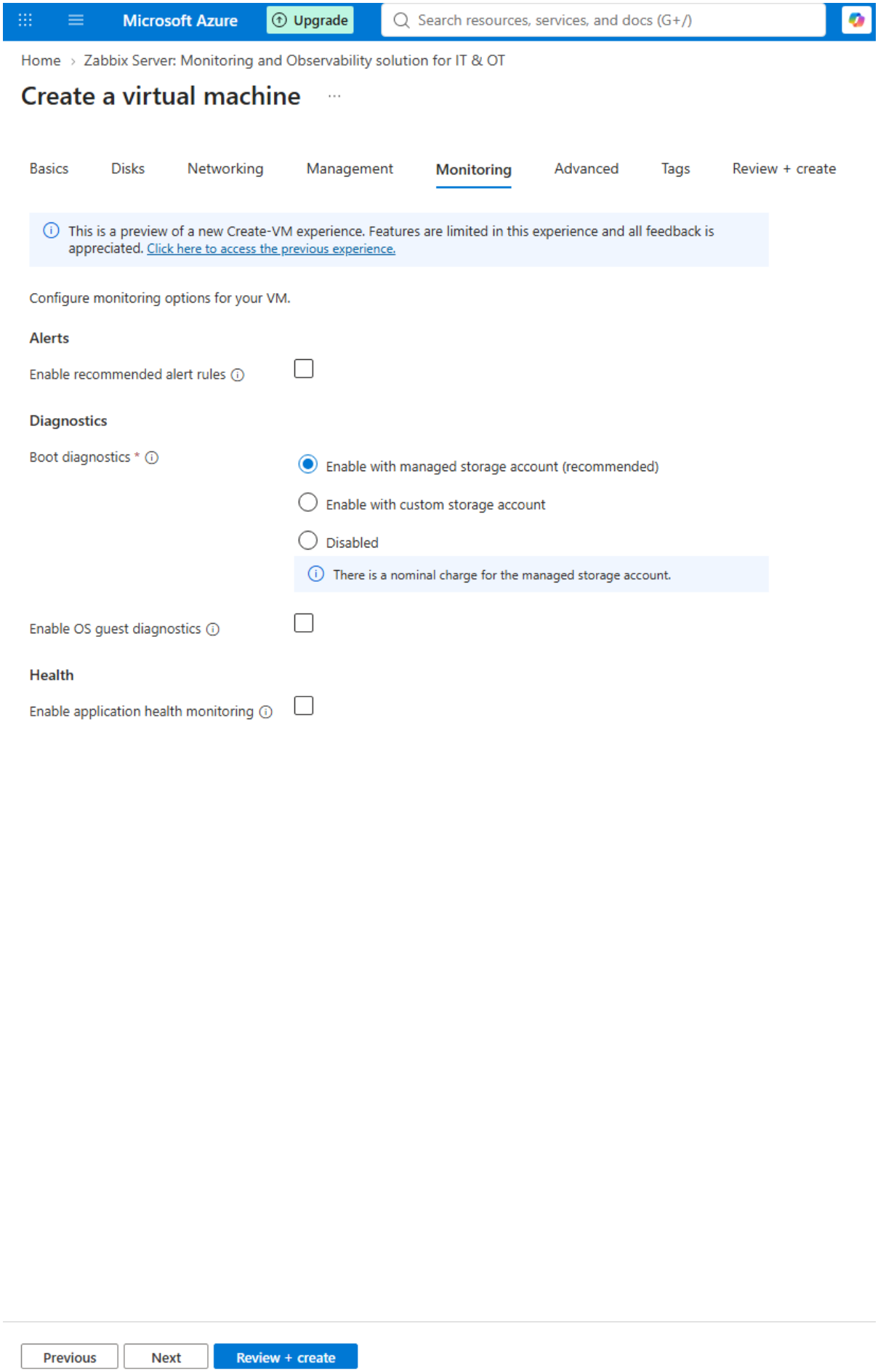
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



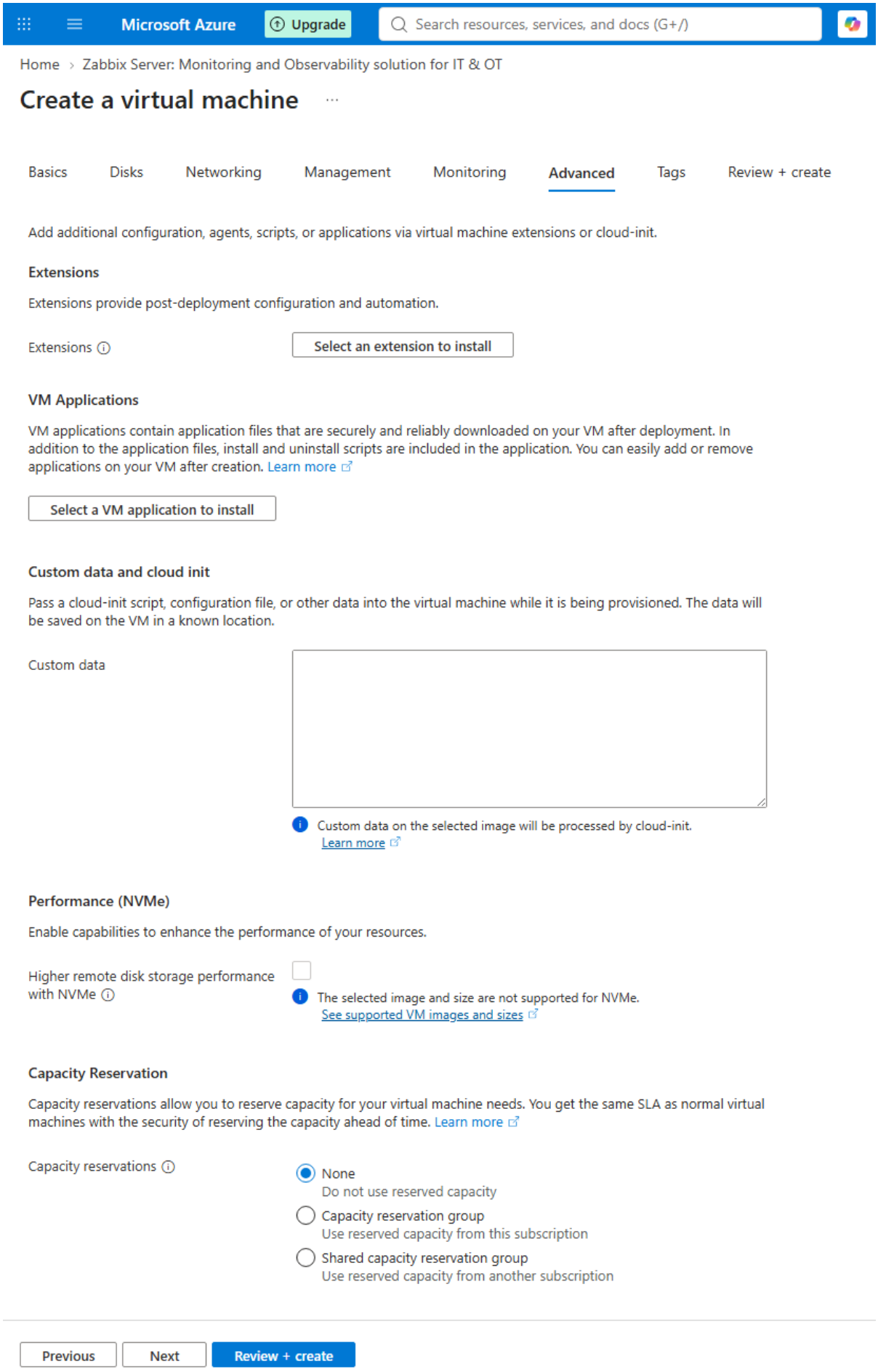
Вкладка *Management*: параметры можно оставить по умолчанию.



Вкладка *Monitoring*: параметры можно оставить по умолчанию.



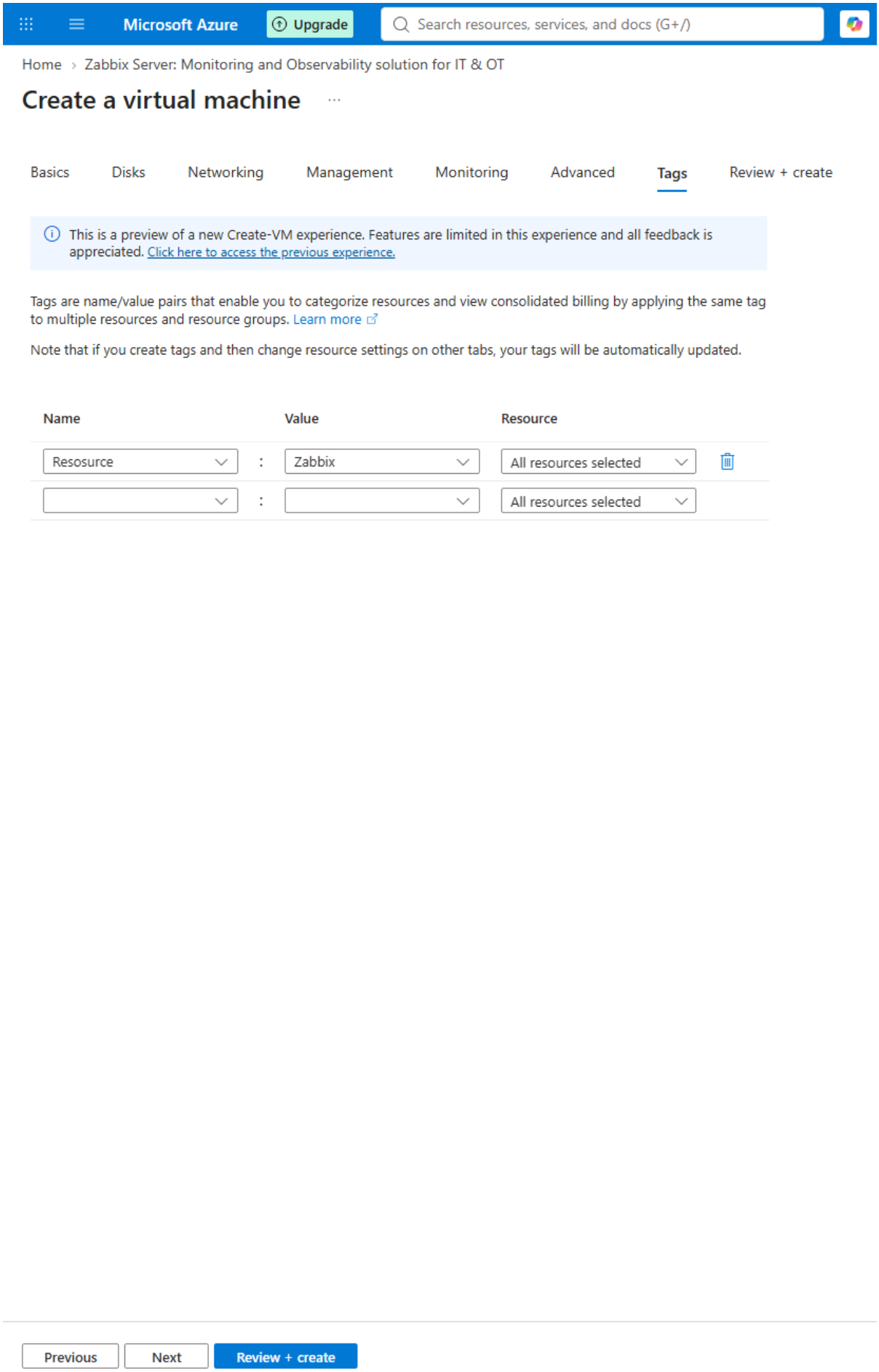
Вкладка *Advanced*: параметры можно оставить по умолчанию.



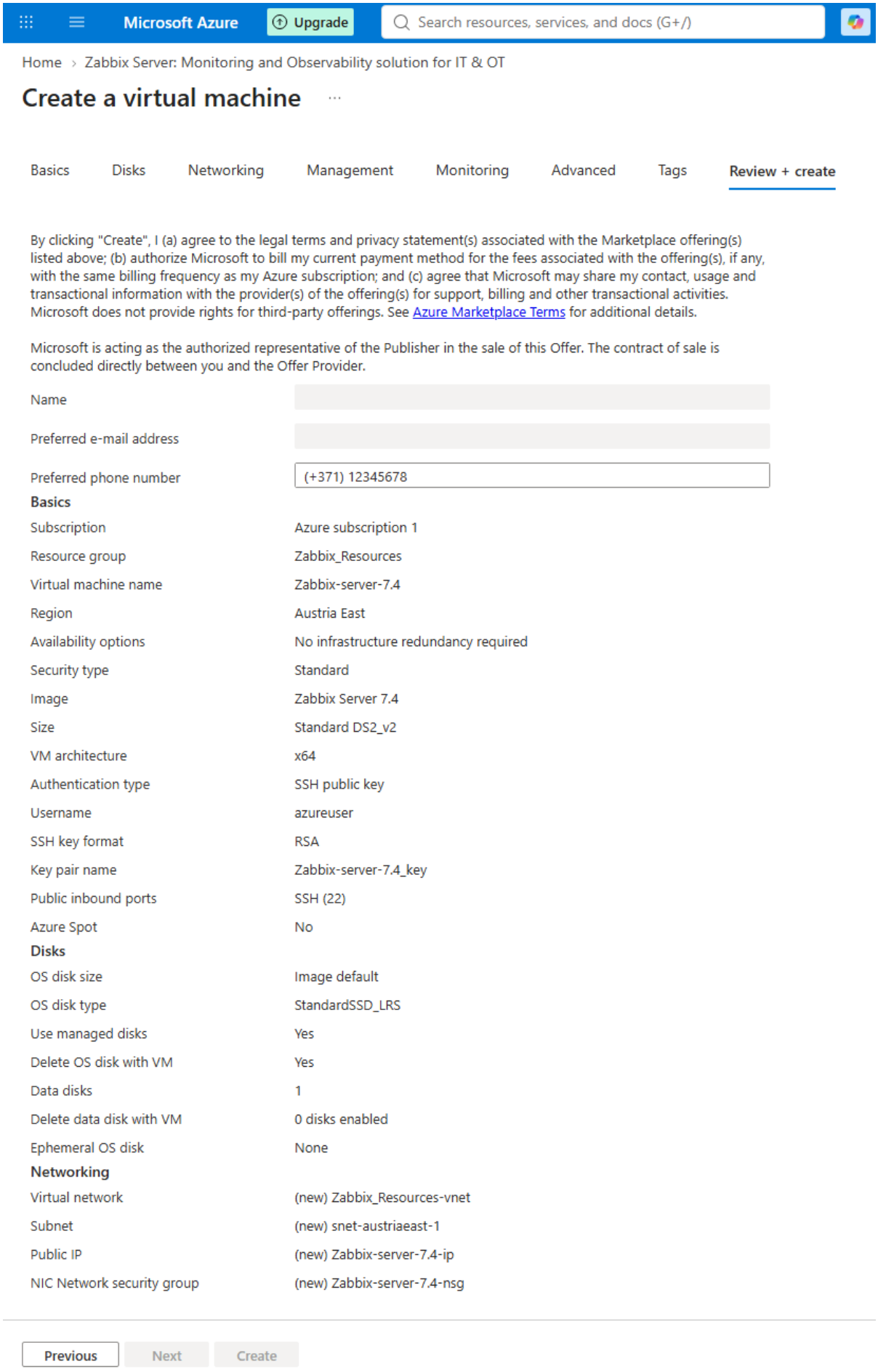
Вкладка *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Проверьте сведения о виртуальной машине и создайте ее.



Вы будете перенаправлены на страницу состояния развертывания виртуальной машины. Дождитесь завершения развертывания.

После завершения развертывания откройте ресурс и запишите его общедоступный IP-адрес. Теперь необходимо

подключиться к серверу Zabbix по SSH (или другим доступным способом), чтобы получить учетные данные для входа в веб-интерфейс Zabbix.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

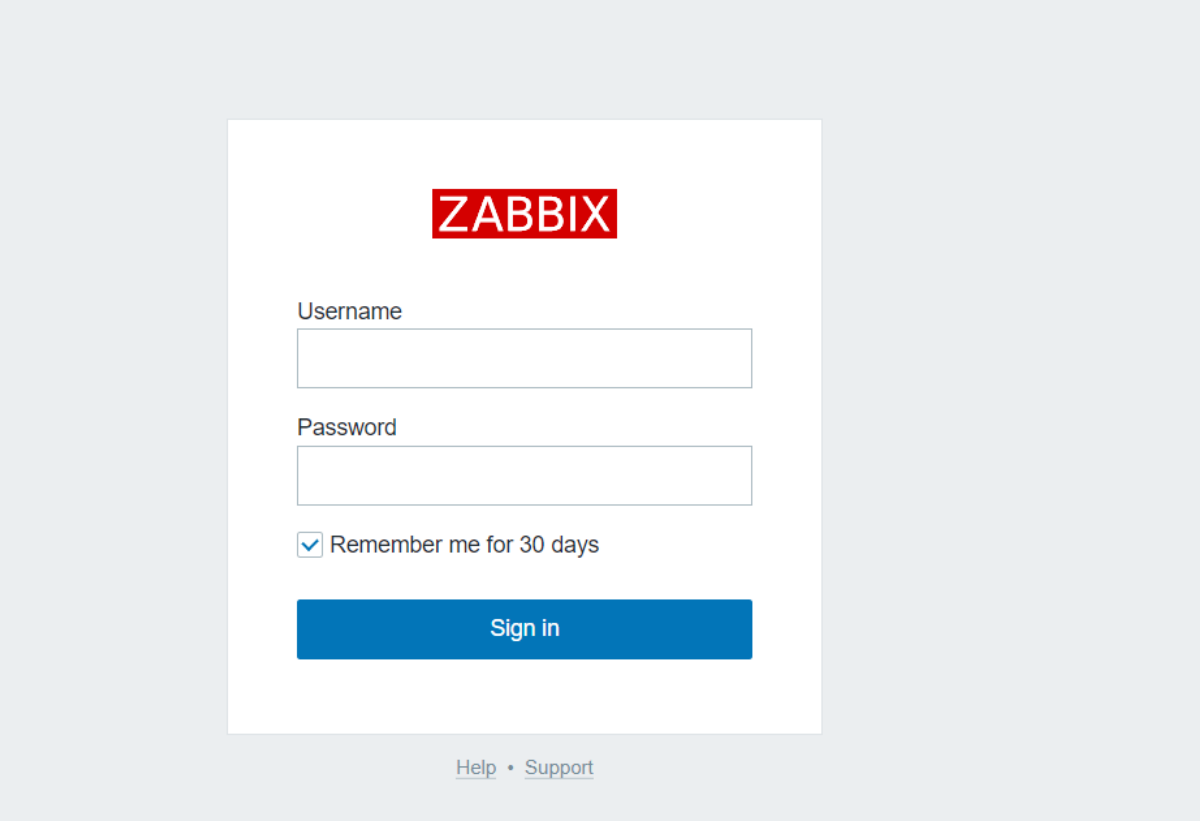

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Откройте веб-интерфейс Zabbix, введя в веб-браузере общедоступный IP-адрес устройства Zabbix. Для входа используйте имя пользователя и пароль, полученные на предыдущем шаге.



Доступ к базе данных

Вы можете получить доступ к базе данных Zabbix с помощью пользователя `root` без ввода пароля. Учетные данные хранятся в файле конфигурации `/root/.my.cnf`.

Руководство по развертыванию прокси Zabbix в Azure

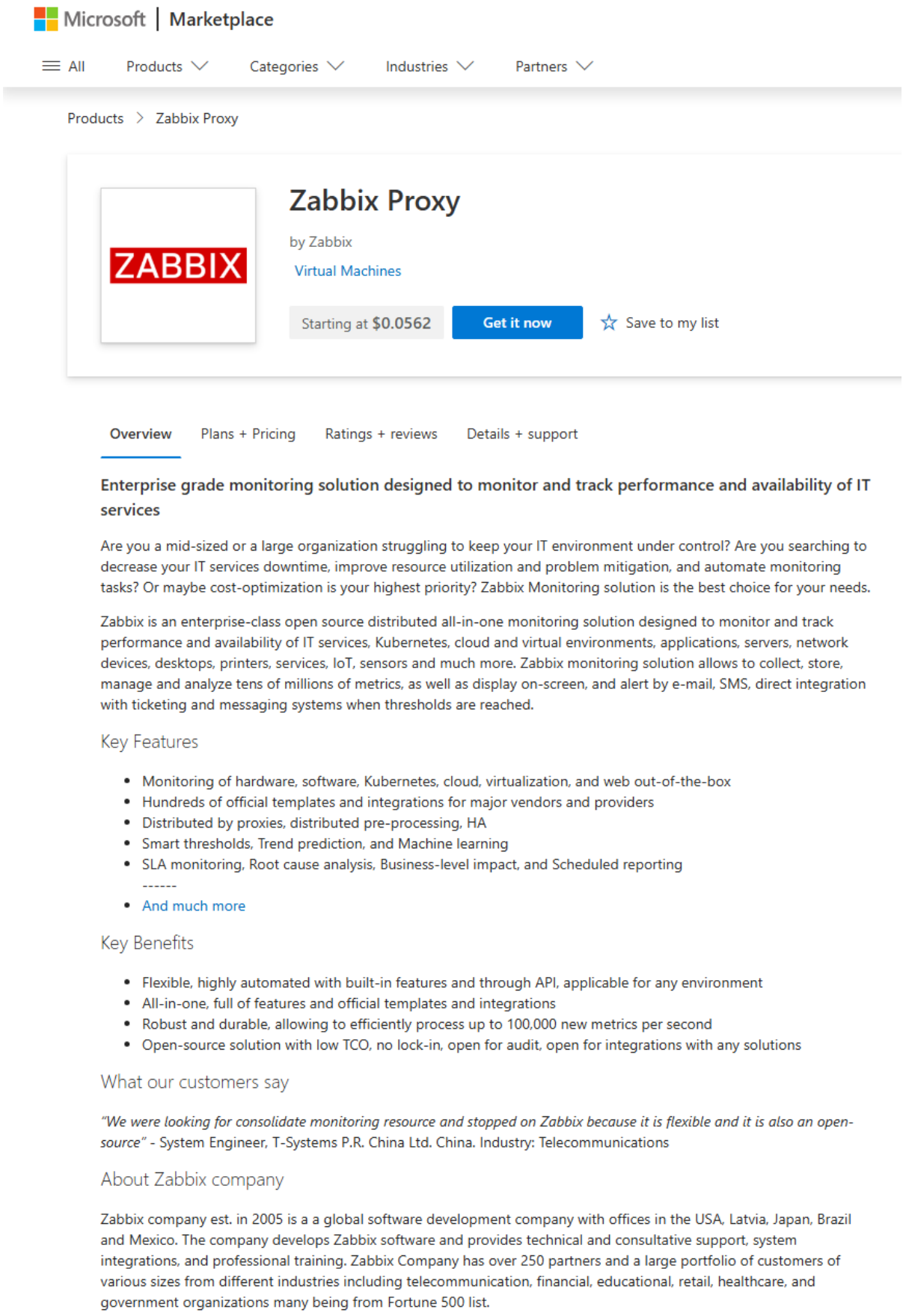
Обзор

На этой странице описывается развертывание прокси Zabbix с использованием Azure.

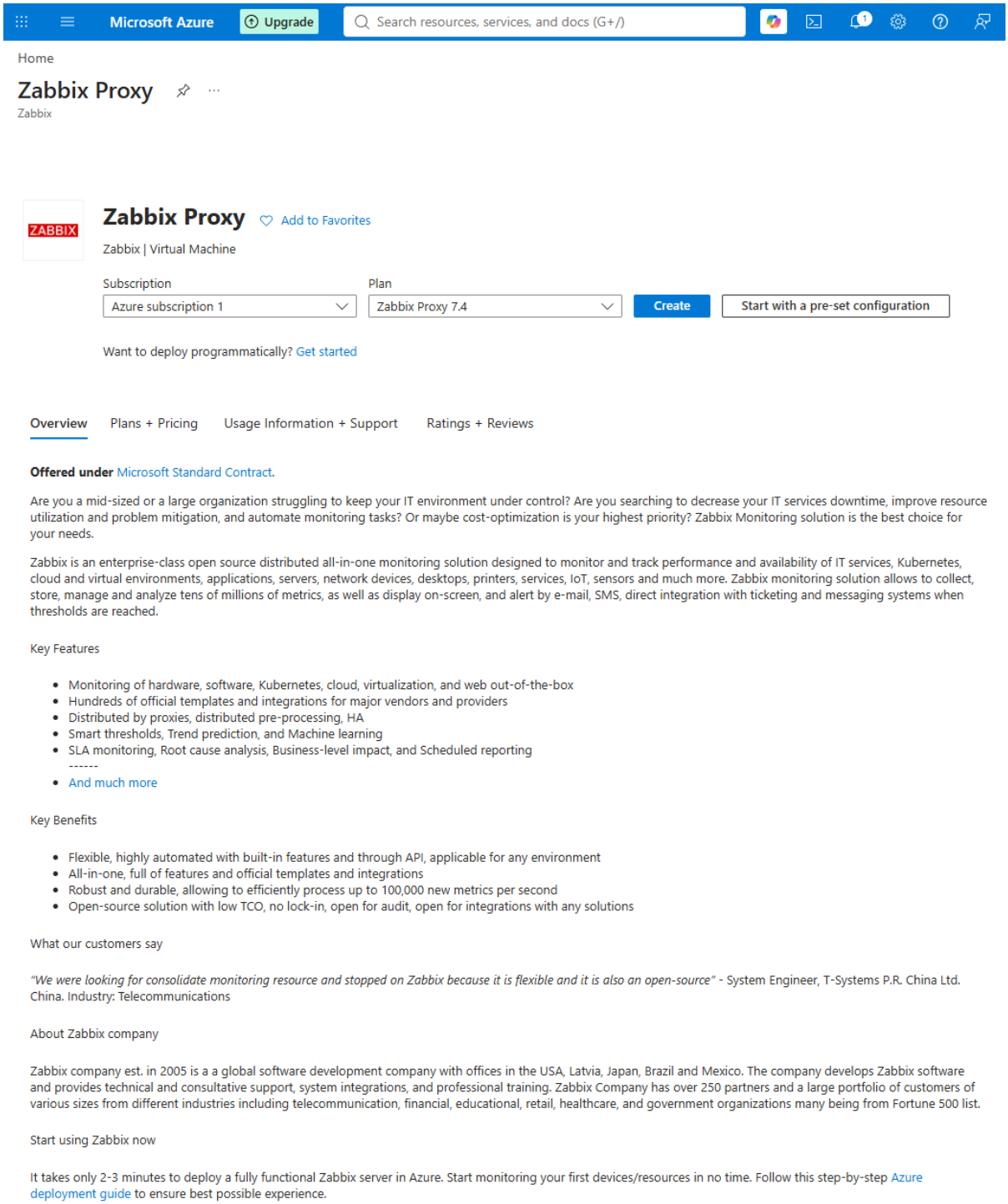
Attention:
Для развертывания на платформе доступны несколько версий Zabbix. Обязательно ознакомьтесь с документацией для той же версии Zabbix, которую вы развернули. Документацию для разных версий Zabbix можно выбрать в раскрывающемся меню версий.

Развертывание прокси Zabbix

Чтобы развернуть прокси Zabbix, сначала перейдите на страницу Zabbix в Azure Marketplace. Нажмите *Get it now*.



Здесь можно ознакомиться с дополнительной информацией о текущем развертывании, например с общим обзором, тарифными планами и другими сведениями. Нажмите *Create* или *Start with a pre-set configuration*.

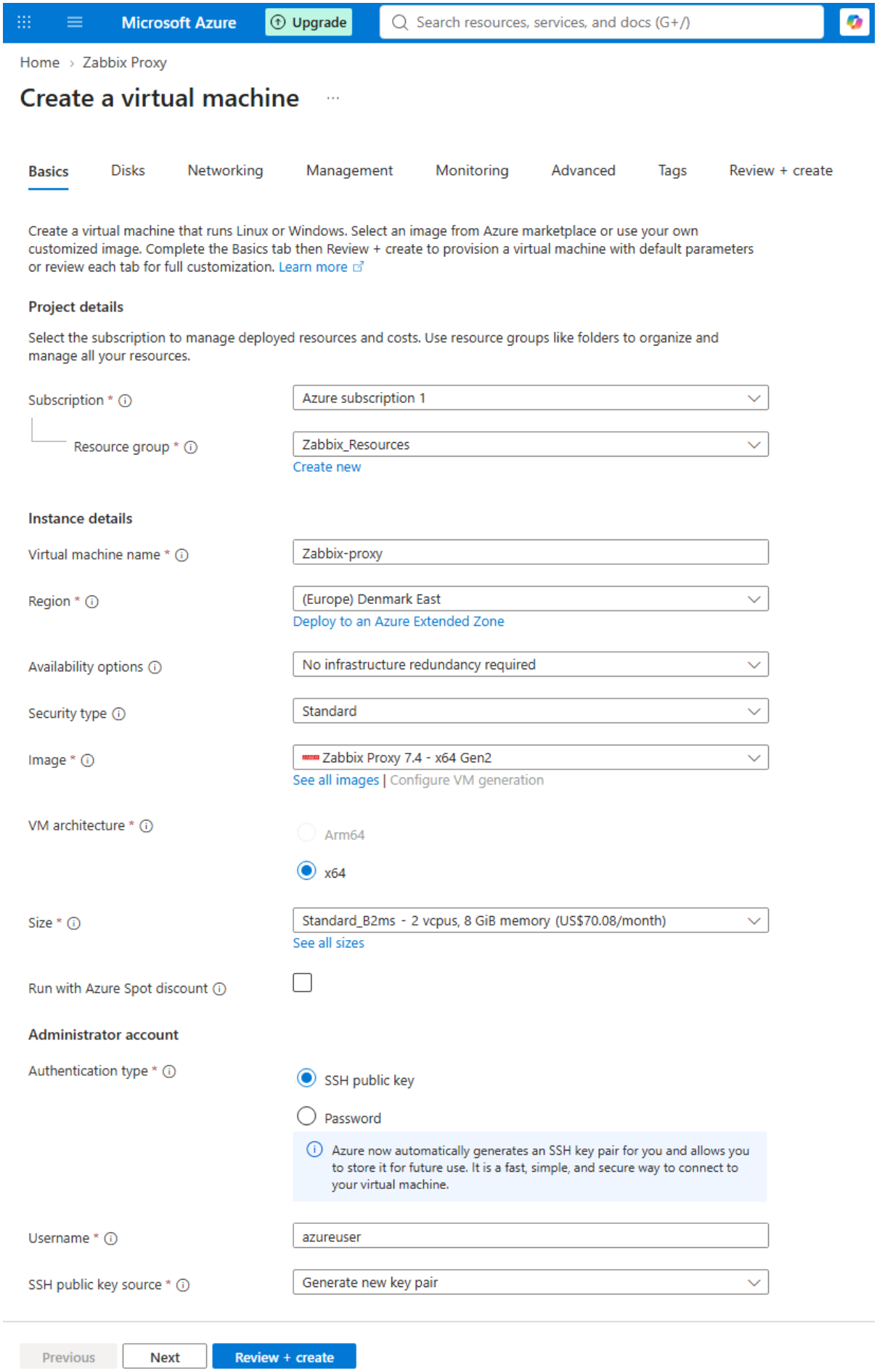


Перейдите к подробной настройке виртуальной машины: выберите размер виртуальной машины, учетную запись администратора, диск, сеть, параметры управления и другие настройки. Эти параметры могут различаться в зависимости от требований вашей инфраструктуры.

В качестве примера можно использовать следующую конфигурацию:

Вкладка *Basics*:

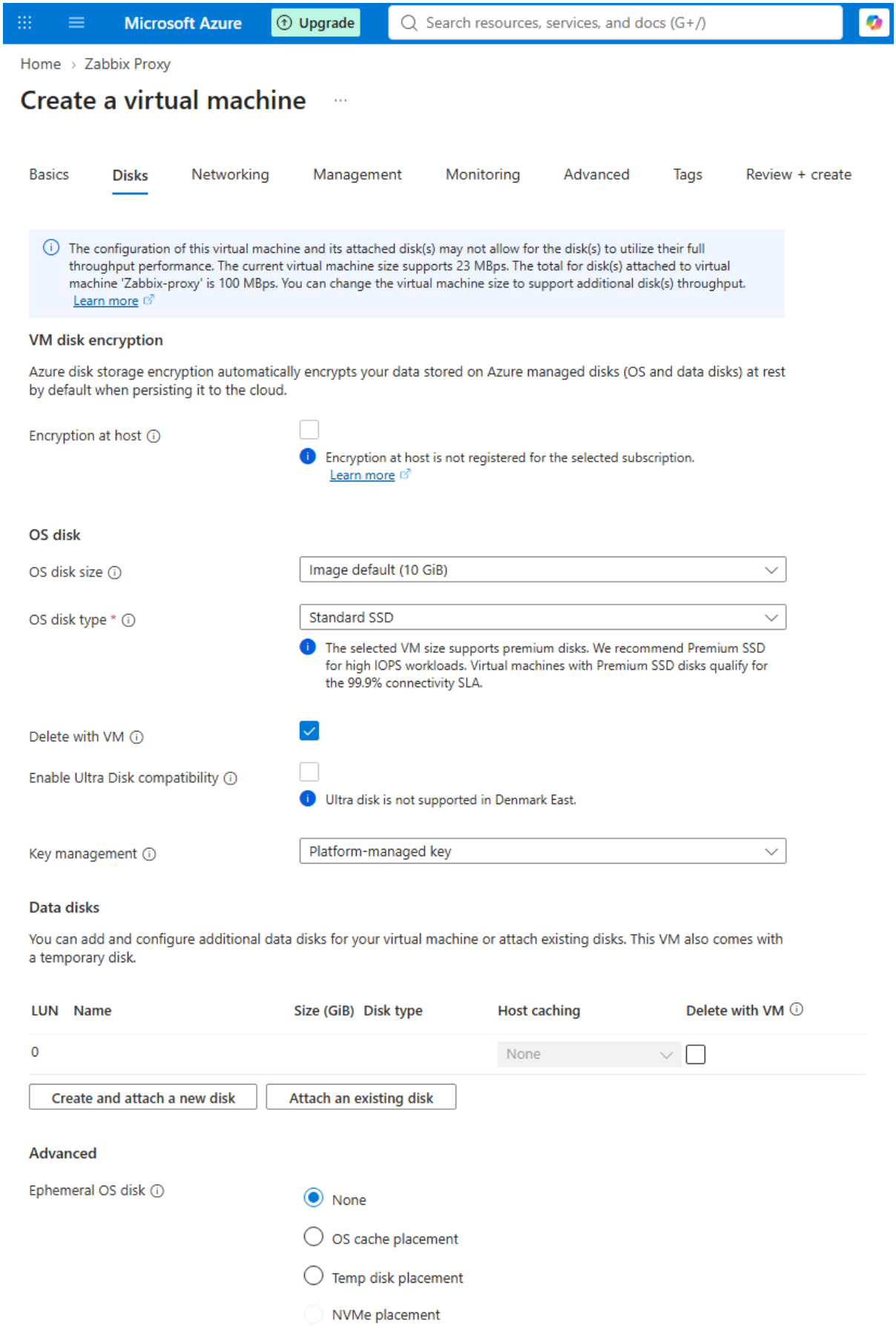
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Вкладка *Disks*:

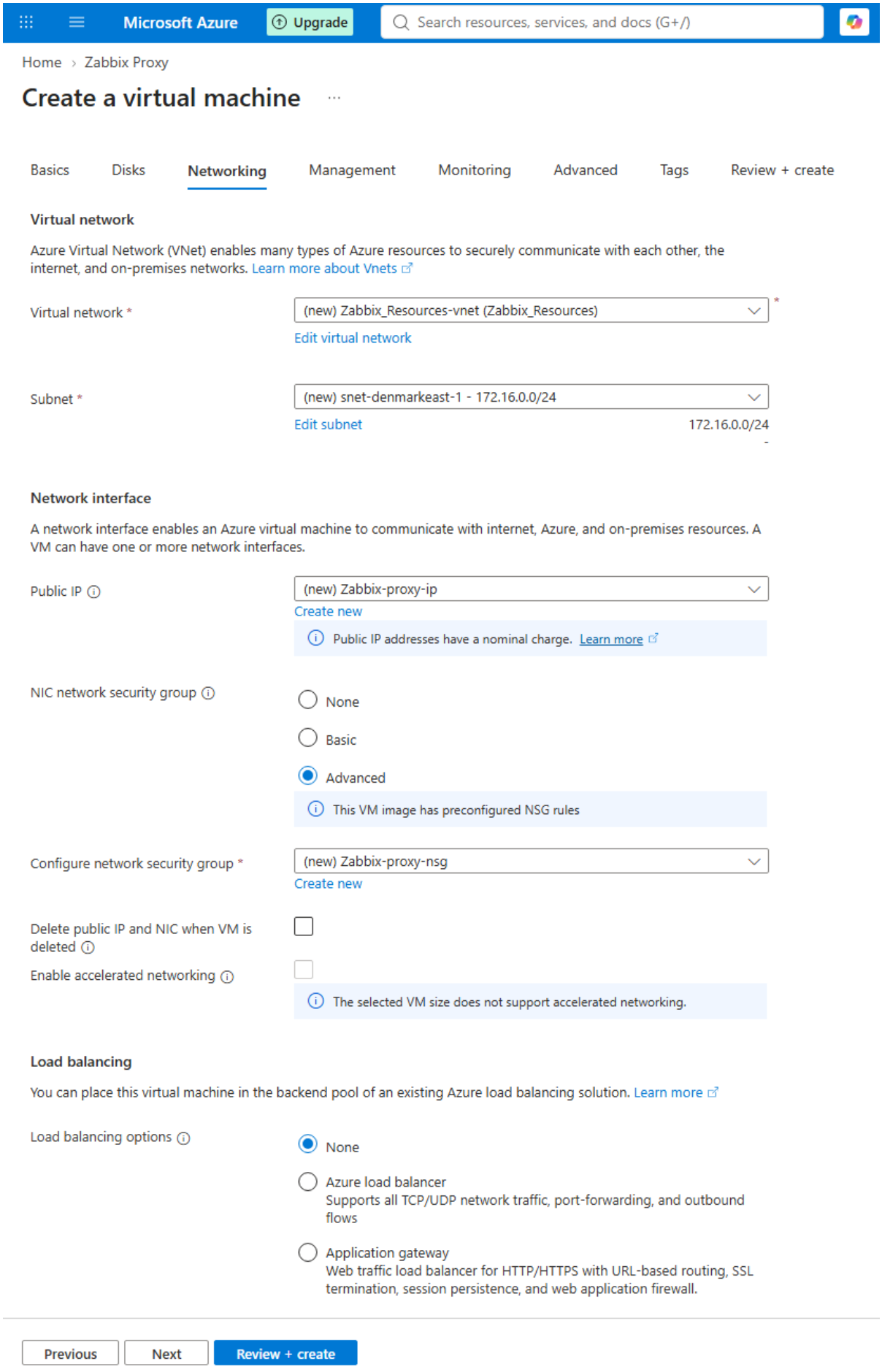
- *Encryption at host*: unchecked

- *OS disk type*: Standard SSD
- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

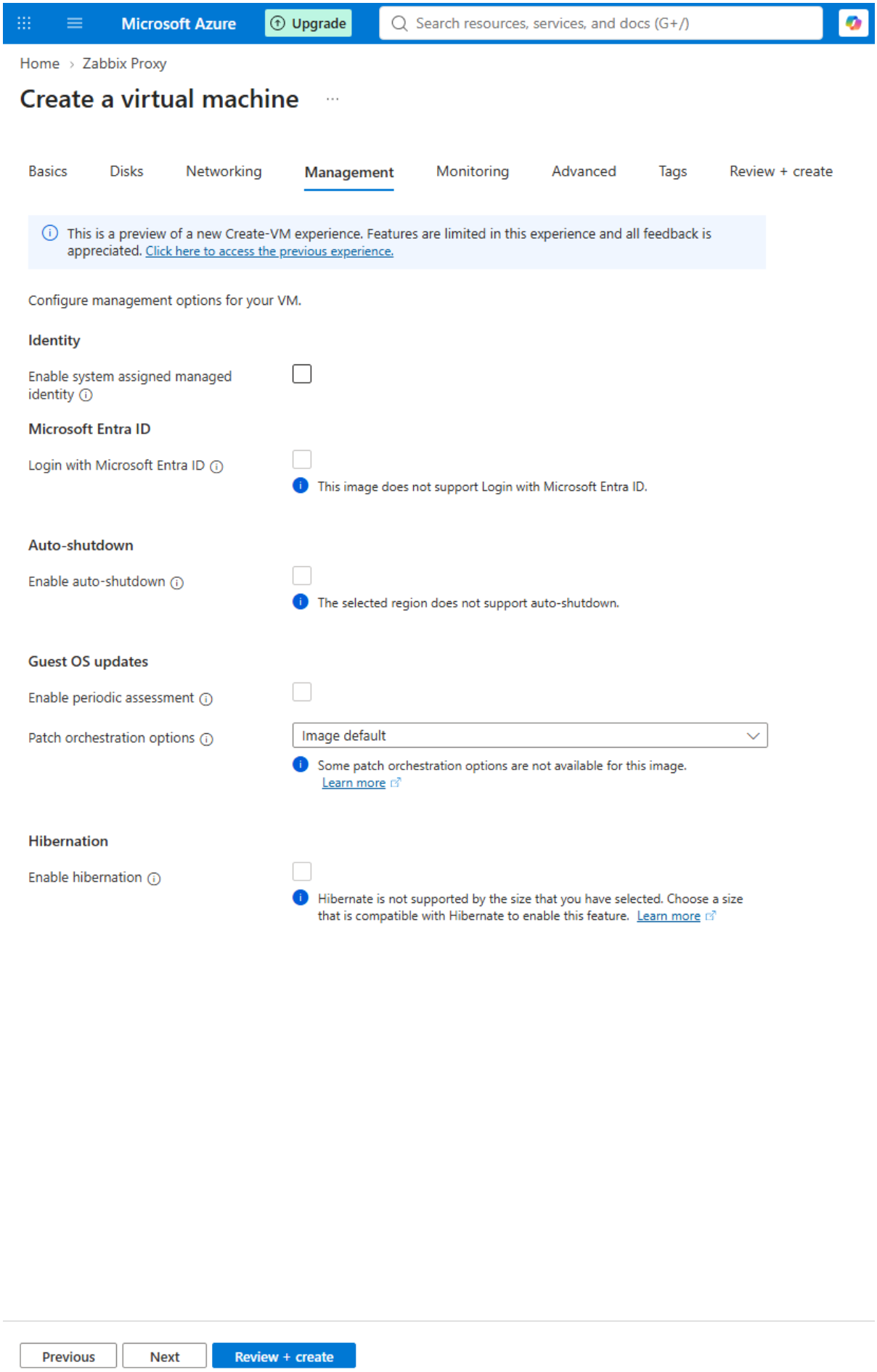


Вкладка *Networking*:

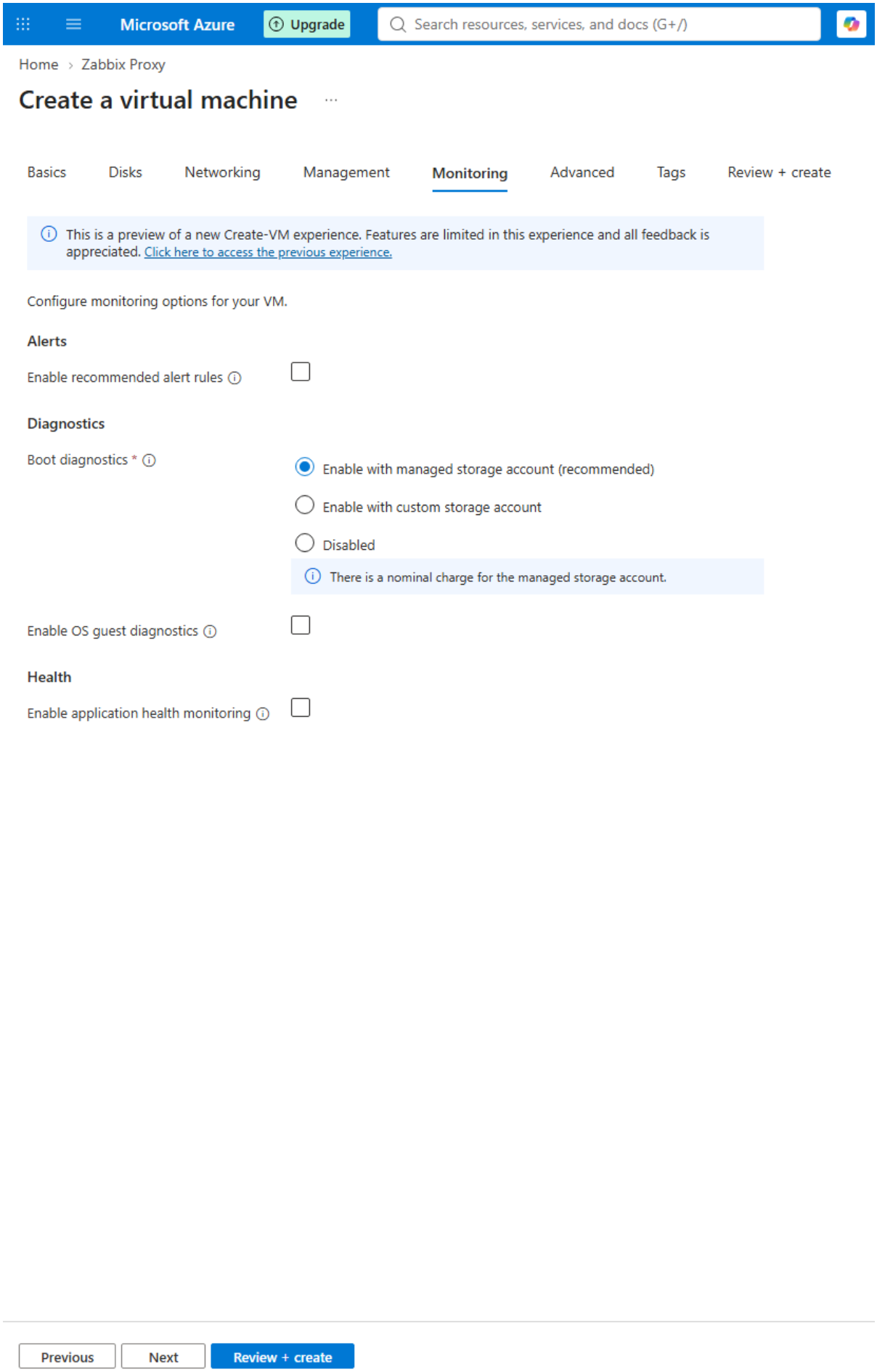
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



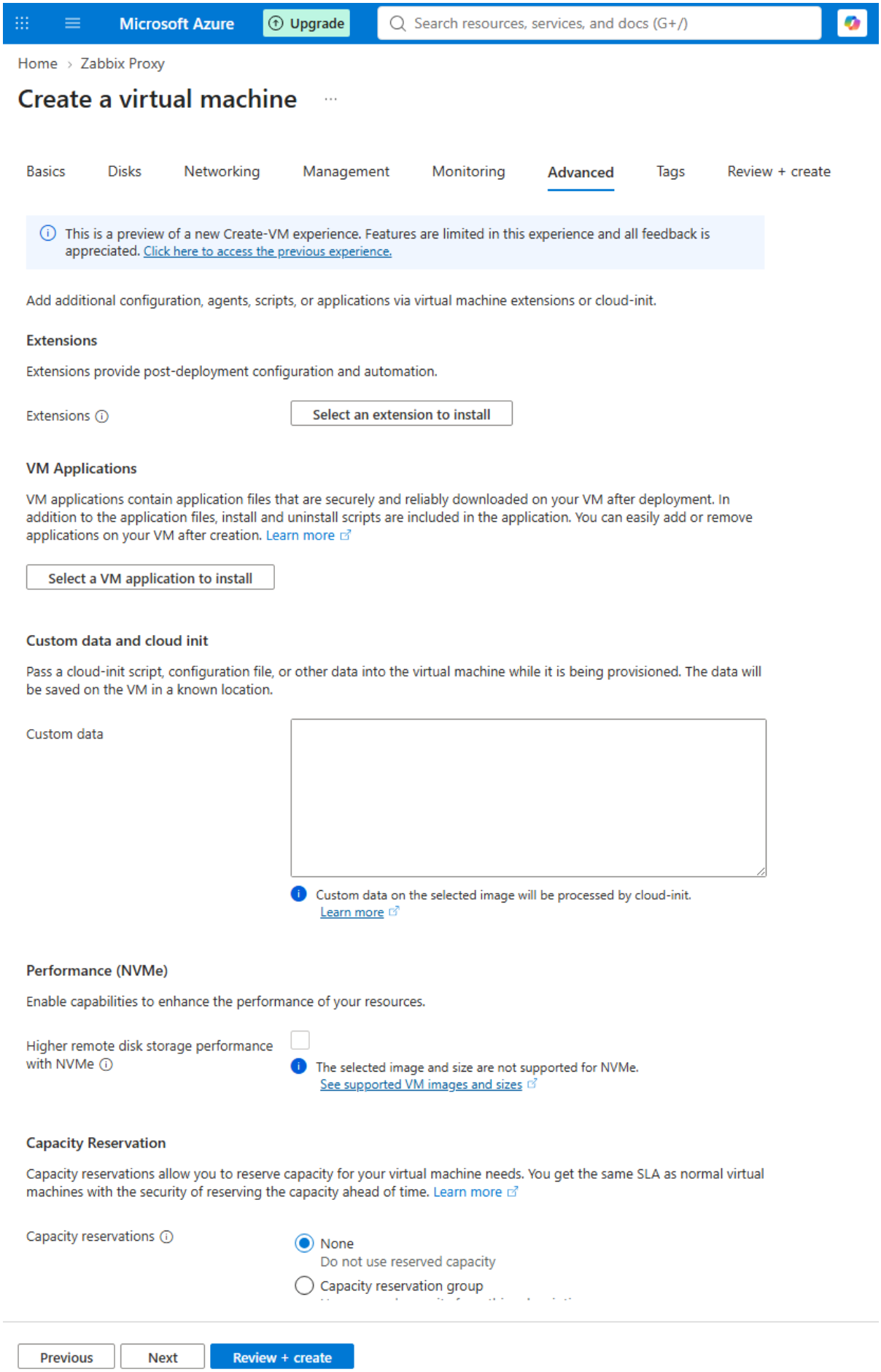
Вкладка *Management*: параметры можно оставить по умолчанию.



Вкладка *Monitoring*: параметры можно оставить по умолчанию.



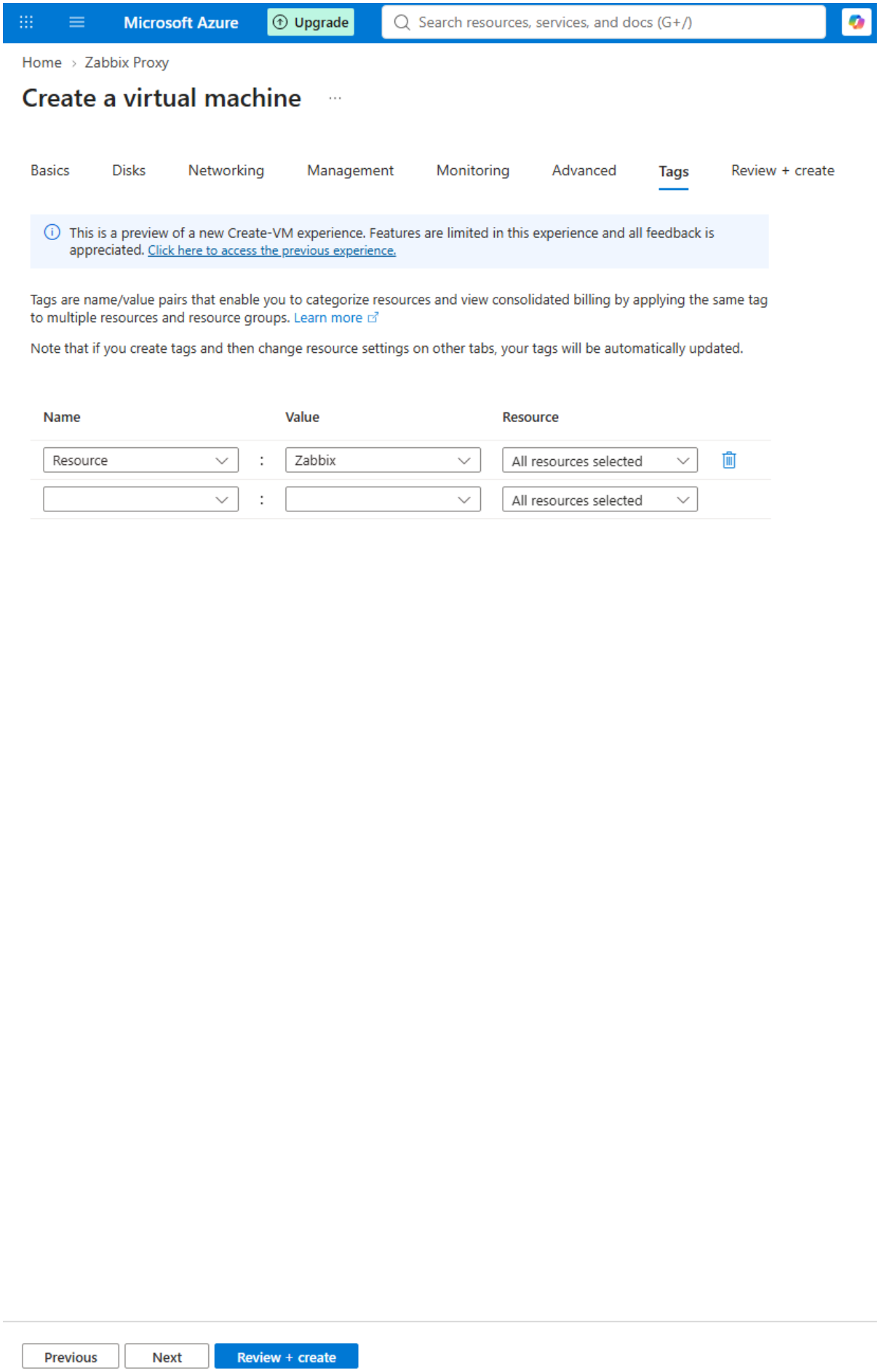
Вкладка *Advanced*: параметры можно оставить по умолчанию.



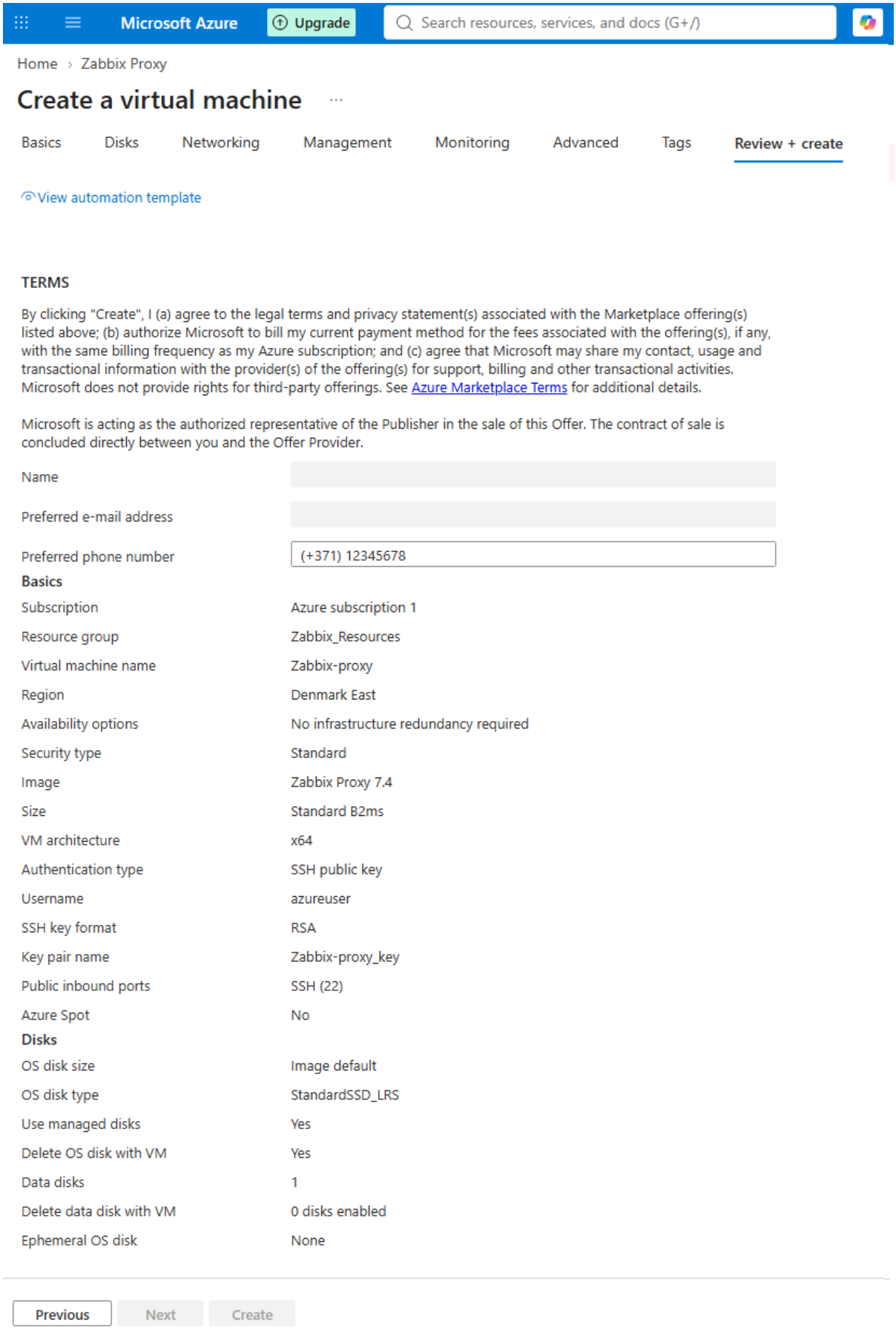
Вкладка *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Проверьте сведения о виртуальной машине и создайте ее.



Вы будете перенаправлены на страницу состояния развертывания виртуальной машины. Дождитесь завершения развертывания.

После завершения развертывания откройте ресурс и запишите его общедоступный IP-адрес. Подключитесь к прокси Zabbix по SSH, используя учетные данные администратора, созданные во время развертывания виртуальной машины, чтобы настроить прокси.

```
*******************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

*******************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Файл конфигурации прокси Zabbix находится в `/etc/zabbix/zabbix_proxy.conf`.

Первоначальная настройка прокси заключается в изменении параметров `ProxyMode`, `Server` и `Hostname` в файле конфигурации прокси Zabbix.

Режим прокси можно изменить, отредактировав параметр `ProxyMode`. Установите `ProxyMode=0` для активного режима или `ProxyMode=1` для пассивного режима.

Параметр `Server` должен содержать IP-адрес сервера Zabbix.

Параметр `Hostname` должен содержать имя узла сети прокси Zabbix. Указанное здесь имя узла сети должно совпадать с именем прокси, заданным при настройке прокси в веб-интерфейсе Zabbix.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Откройте веб-интерфейс Zabbix и перейдите в раздел *Administration* > *Proxies*. Нажмите кнопку *Create proxy*.

Введите то же имя прокси, которое указано в качестве значения параметра *Hostname* в файле конфигурации прокси.

Выберите режим прокси и укажите IP-адрес прокси (необязательно для активных прокси).

Нажмите кнопку *Create proxy*, чтобы создать прокси.

Расположение базы данных прокси

Базу данных SQLite прокси можно найти в `/tmp/zabbix_proxy.sqlite3`; она задается параметром `DBName` в файле конфигурации прокси Zabbix.

Руководство по развертыванию в Google Cloud

Обзор

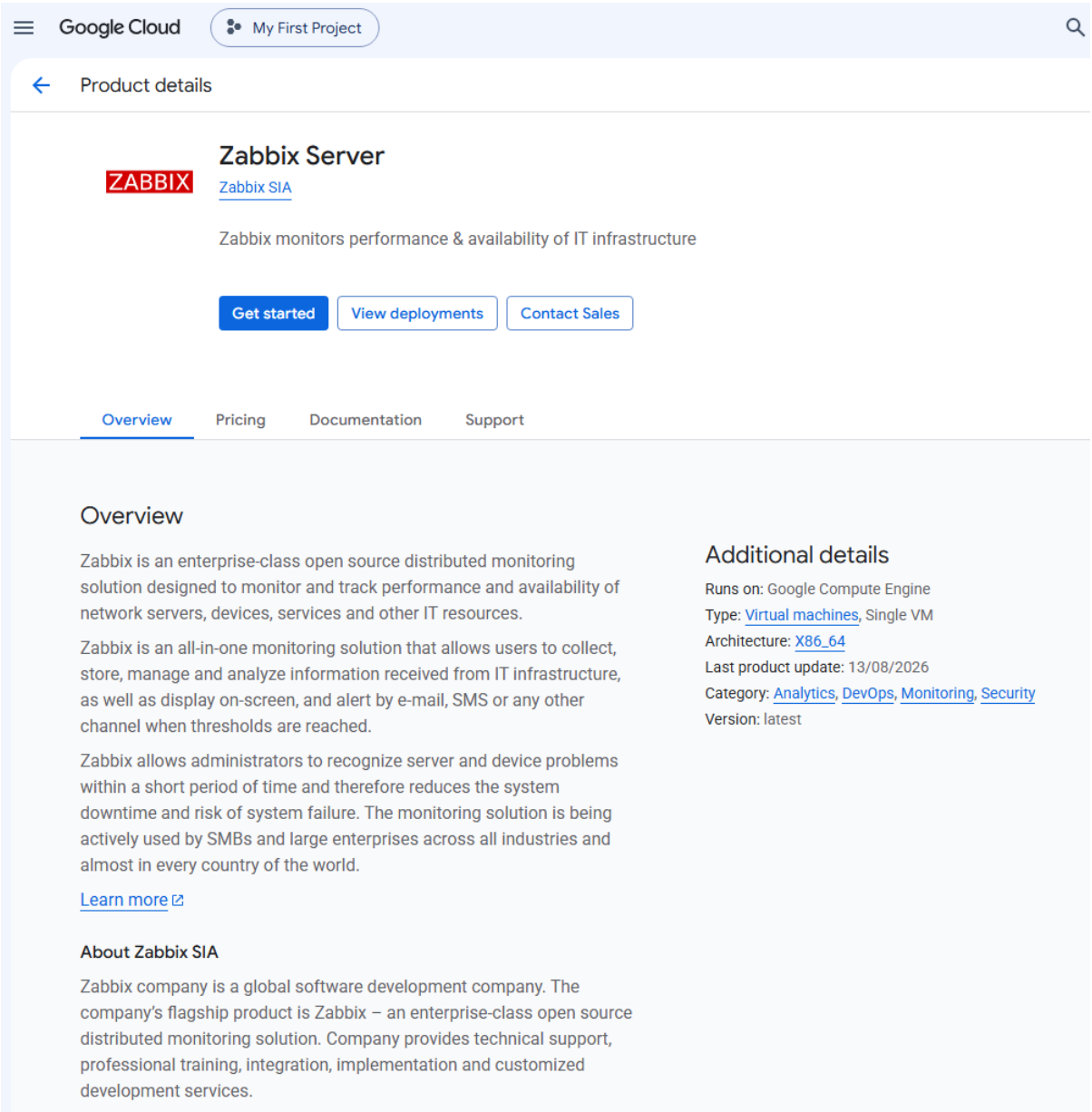
На этой странице описывается развертывание Zabbix с помощью Google Cloud Marketplace.

Attention:
На платформе доступны для развертывания несколько версий Zabbix. Обязательно ознакомьтесь с документацией для той же версии Zabbix, которую вы развернули. Документацию для разных версий Zabbix можно выбрать в раскрывающемся меню версий.

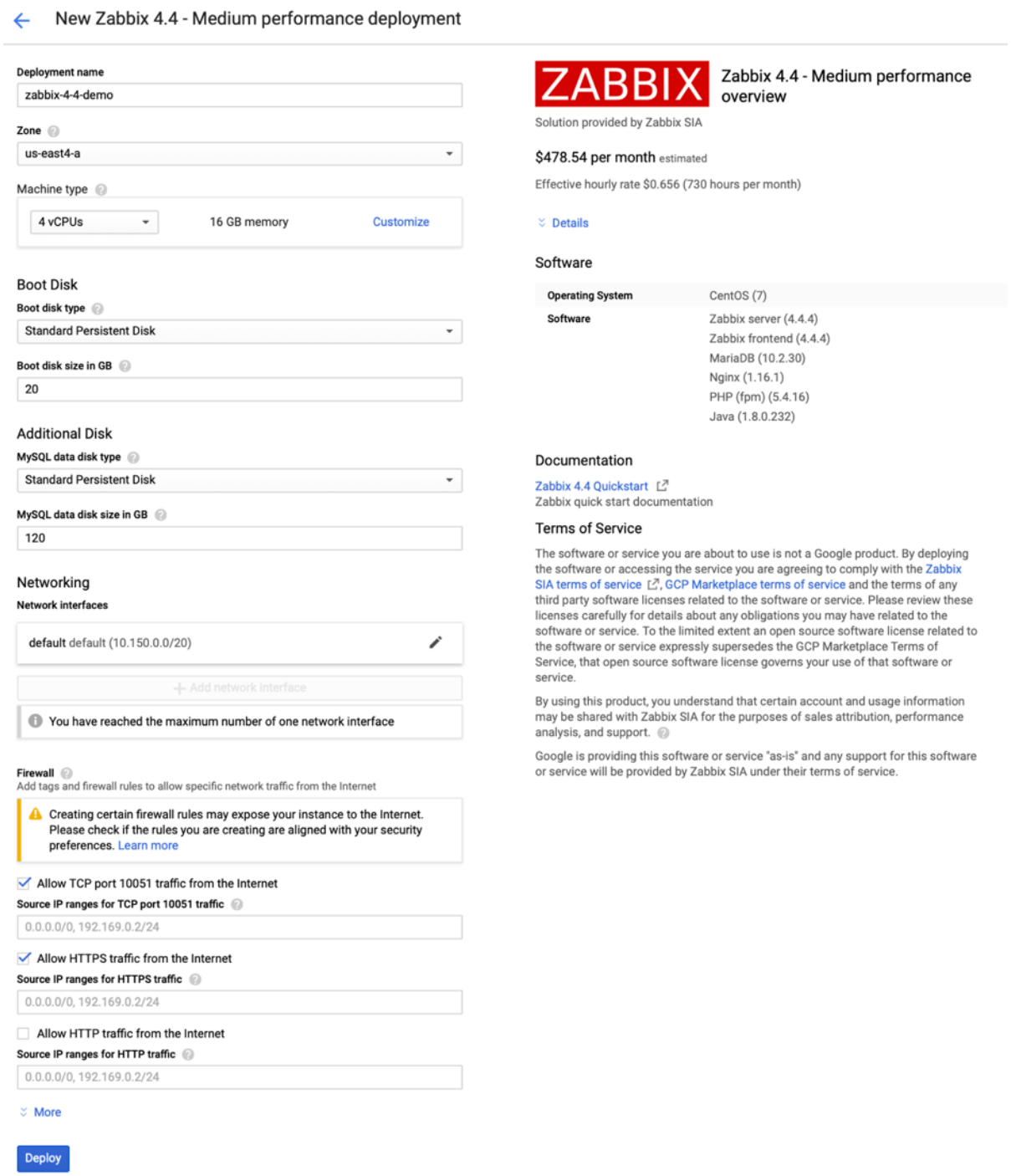
Развертывание устройства

Сначала разверните устройство Zabbix. Сделать это очень просто с помощью Google Cloud Marketplace. Просто перейдите к продукту Zabbix Server на торговой площадке.

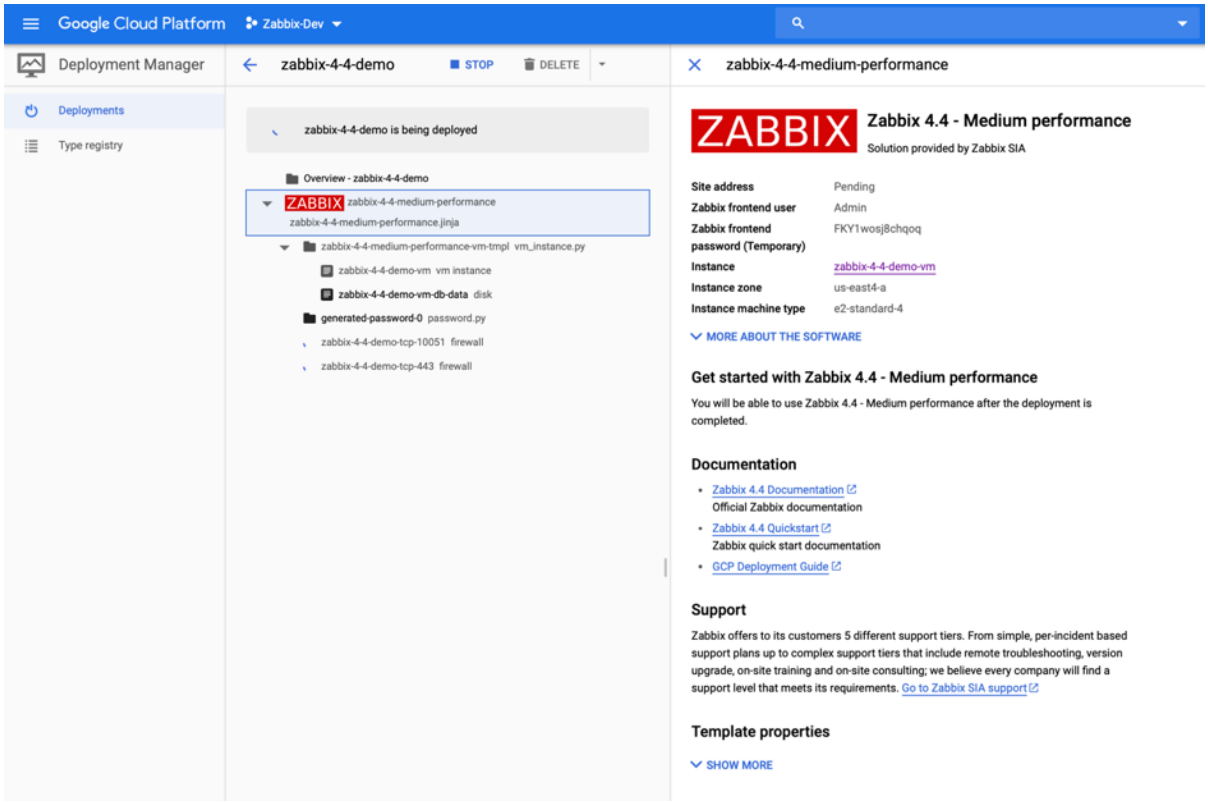
Нажмите *Get started*.



Можно оставить настройки по умолчанию или изменить их. После завершения нажмите *Deploy*.



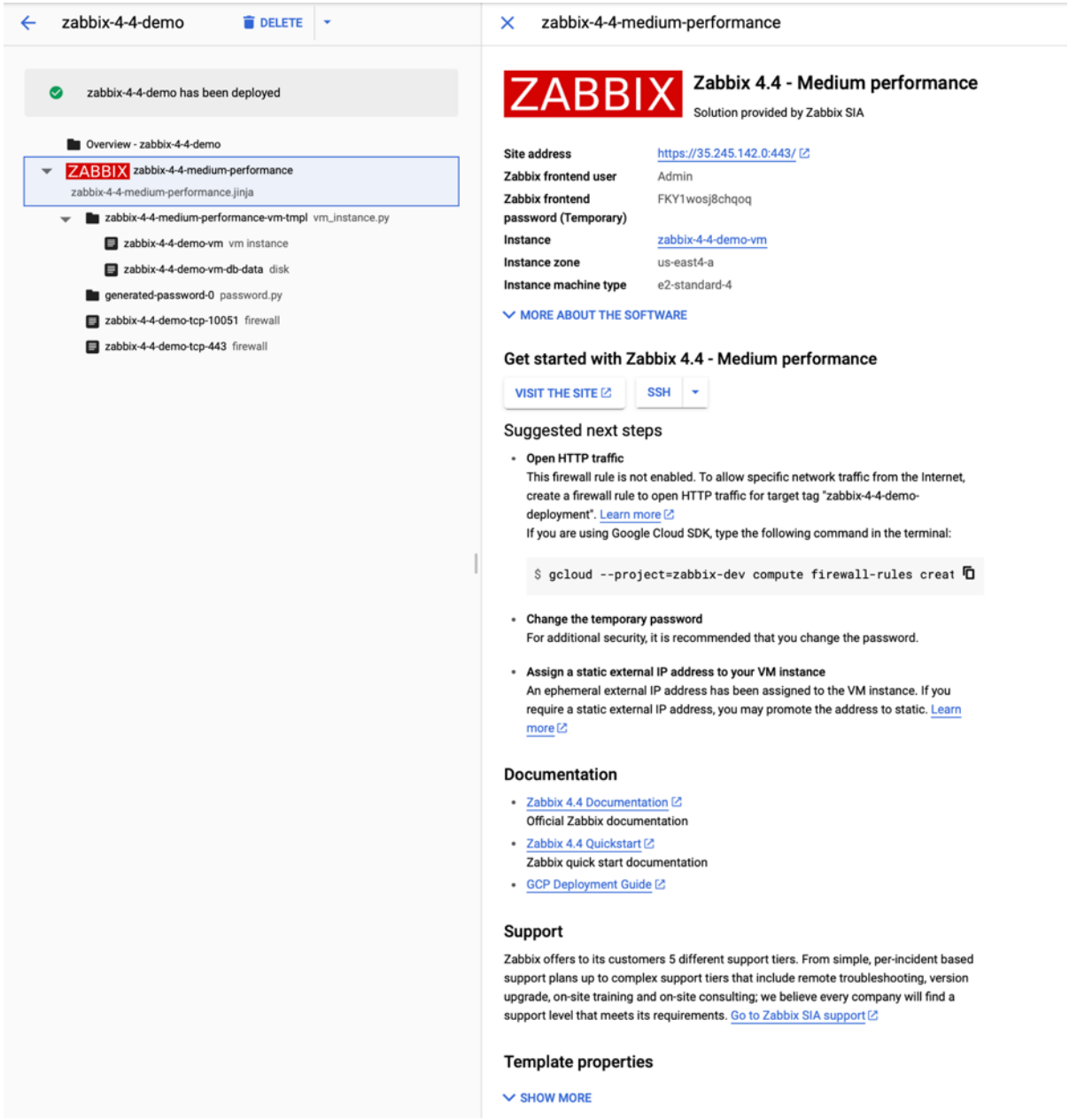
Теперь выполняется развертывание. Дождитесь его завершения.



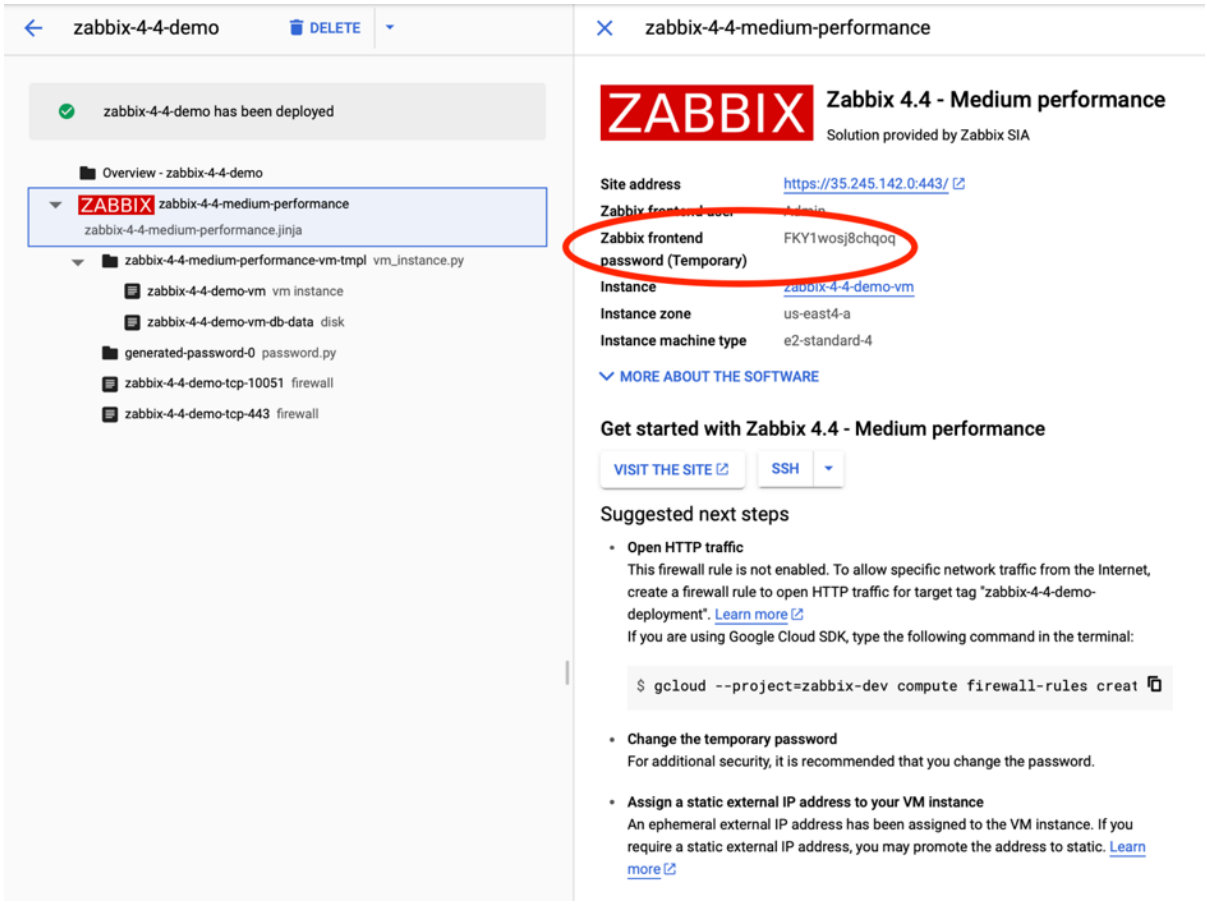
Готово! Развертывание началось.

Доступ к устройству

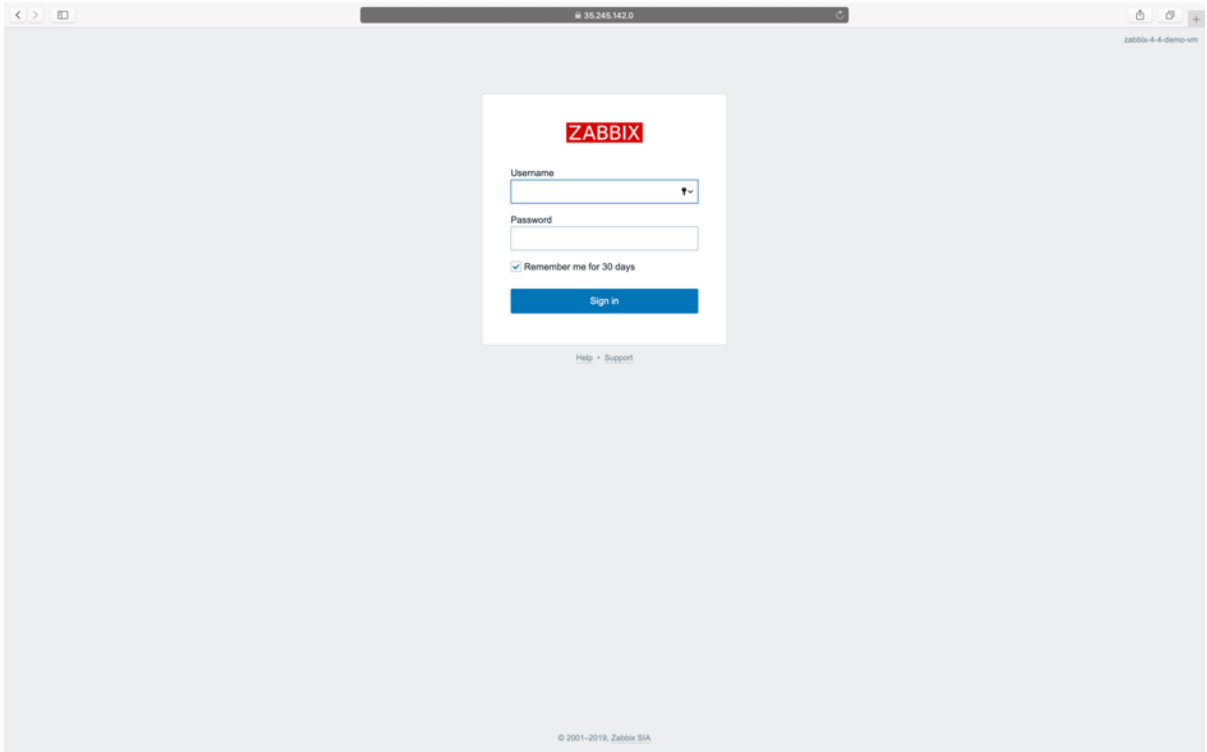
После завершения развертывания должно появиться сообщение о том, что устройство развертывается.

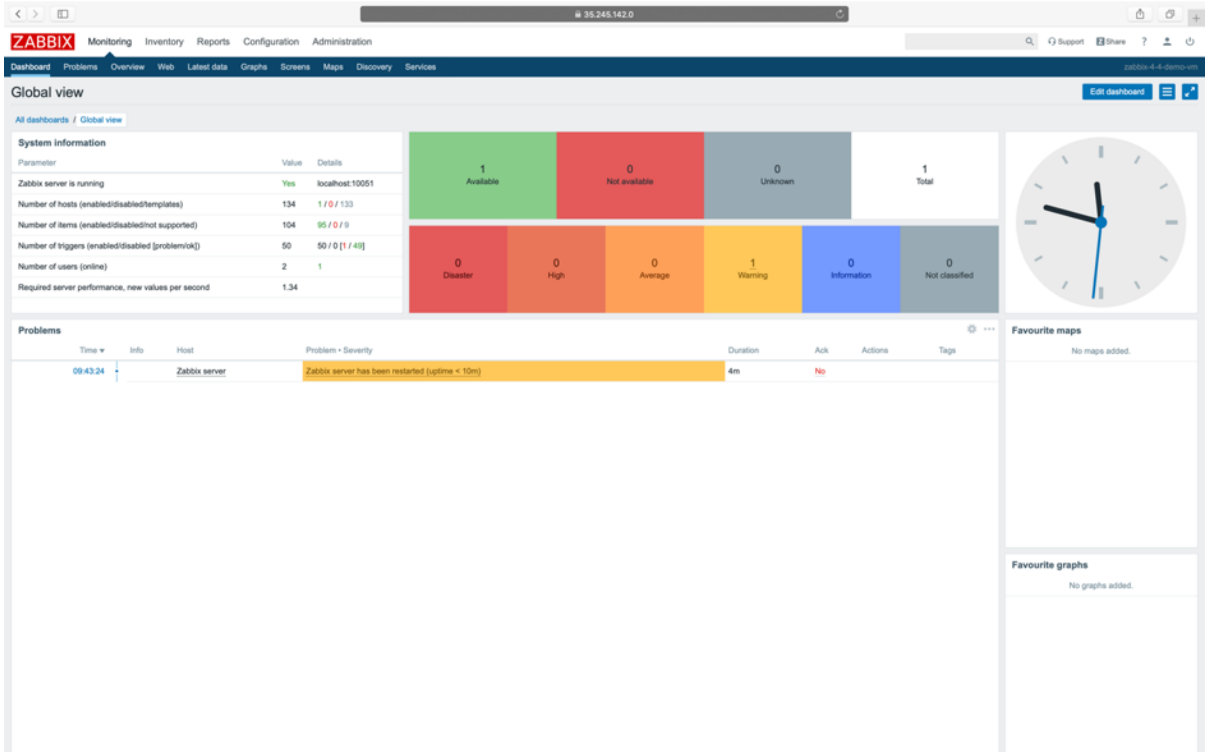


Откройте веб-браузер и перейдите по адресу *Site Address*, чтобы открыть веб-интерфейс Zabbix. Перед этим получите пароль пользователя Zabbix *Admin*, чтобы войти в устройство Zabbix.

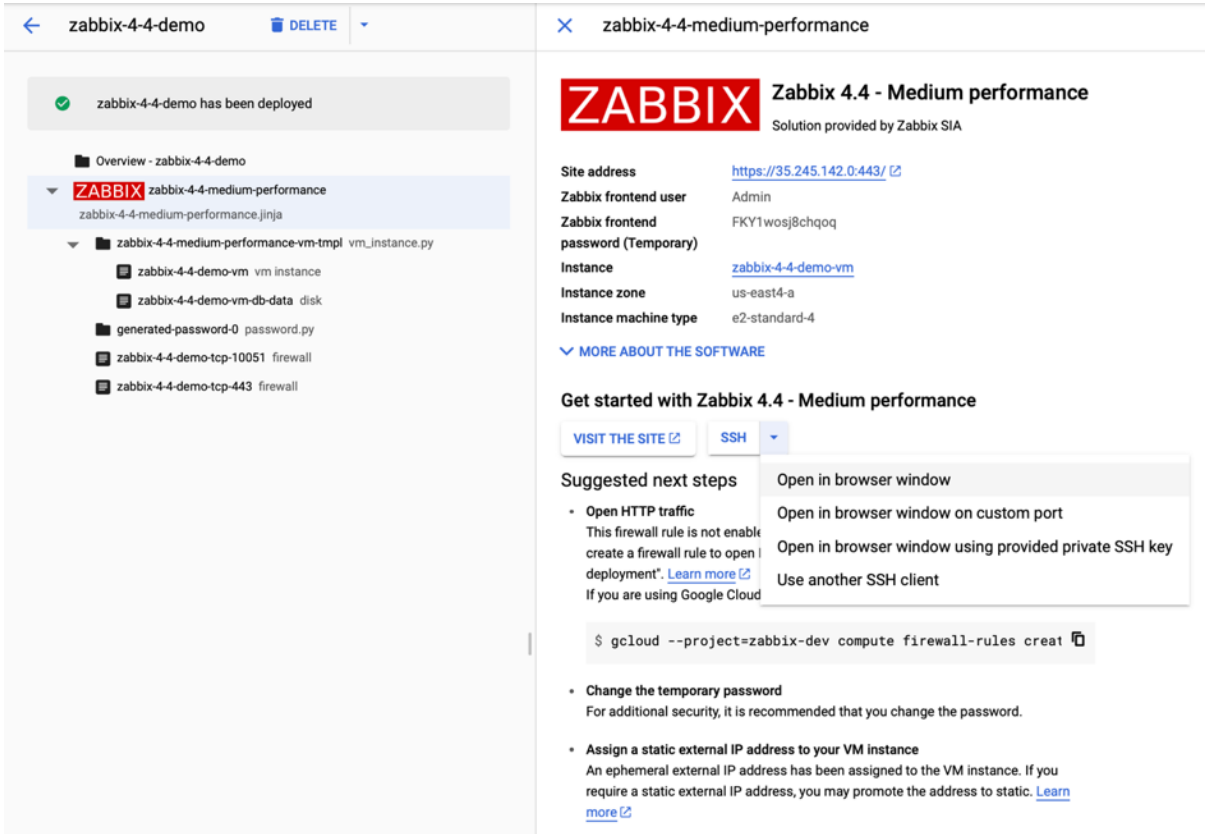


Готово! Теперь у вас есть устройство Zabbix.





Также можно войти в экземпляр, чтобы взаимодействовать с базой данных. Для этого вернитесь в консоль Google и выполните действия, указанные красной стрелкой ниже, чтобы запустить сеанс SSH с помощью параметра *Open in browser window*.



MySQL доступен без пароля `root`. Пароль хранится в файле конфигурации `/root/.my.cnf`. По умолчанию устройство Zabbix использует часовой пояс UTC. Если его необходимо изменить, отредактируйте директиву конфигурации PHP `date.timezone` в файле `/etc/php.d/99-zabbix.ini` и перезапустите процесс php-fpm: `systemctl restart php-fpm`.

Обновление

Перед обновлением настоятельно рекомендуется ознакомиться со всеми соответствующими примечаниями к обновлению.

Также будет полезно узнать, какие компоненты необходимо обновить.

Обзор

Пошаговые инструкции по обновлению доступны для:

- Red Hat Enterprise Linux (с использованием пакетов)
- Debian/Ubuntu (с использованием пакетов)
- Обновление с использованием контейнеров
- Обновление с использованием исходного кода

Связанные инструкции:

- Для серверов в кластере высокой доступности (HA) см. Обновление кластера HA
- Для базы данных TimescaleDB см. Обновление схемы TimescaleDB

Компоненты для обновления

Обновление агентов Zabbix рекомендуется, но не является обязательным.

Настоятельно рекомендуется обновить прокси Zabbix. Сервер Zabbix полностью поддерживает прокси той же основной версии, что и сервер. Сервер Zabbix также поддерживает прокси, которые **не старше** предыдущей версии Zabbix с долгосрочной поддержкой (LTS), но с ограниченной функциональностью (сбор данных, выполнение удаленных команд, немедленная проверка значений элементов данных). Обновление конфигурации также отключено, и устаревшие прокси будут работать только со старой конфигурацией.

Attention:
Прокси, версия которых старше предыдущей версии LTS сервера Zabbix или новее основной версии сервера Zabbix, не поддерживаются. Сервер Zabbix будет игнорировать данные от неподдерживаемых прокси, и весь обмен данными с сервером Zabbix завершится с предупреждением. Дополнительную информацию см. в разделе Совместимость версий.

Чтобы свести к минимуму время простоя и потерю данных во время обновления, рекомендуется остановить, обновить и запустить сервер Zabbix, а затем поочерёдно останавливать, обновлять и запускать прокси Zabbix. Во время простоя сервера работающие прокси продолжат сбор данных. После запуска сервера устаревшие прокси отправят данные на более новый сервер (при этом конфигурация прокси обновлена не будет) и останутся частично работоспособными. Все уведомления о проблемах, возникших во время простоя сервера Zabbix, будут созданы только после запуска обновлённого сервера.

Если прокси Zabbix запускается впервые и файл базы данных SQLite отсутствует, прокси автоматически создает его.

Обратите внимание, что если прокси Zabbix использует SQLite3 и при запуске обнаруживает, что версия существующего файла базы данных старше требуемой, он **автоматически удалит файл базы данных** и создаст новый. Поэтому данные истории, хранящиеся в файле базы данных SQLite, будут потеряны. Если версия прокси Zabbix старше версии файла базы данных, Zabbix зарегистрирует ошибку и завершит работу.

В зависимости от размера базы данных обновление базы данных до версии 8.0 может занять много времени.

Примечания по обновлению

Прямое обновление до Zabbix 8.0.x поддерживается с версий **2.0**.x. Для обновления с более ранних версий обратитесь к документации Zabbix для версии 2.0 и более ранних версий.

Note:
Обратите внимание, что после обновления некоторые интеграции Zabbix со сторонним программным обеспечением могут быть затронуты, если внешнее программное обеспечение несовместимо с обновленной версией Zabbix.

Доступны следующие примечания к обновлению:

Обновление с версии	Полные примечания к обновлению	Наиболее важные изменения между версиями
7.4.x	Для: Zabbix 8.0	Агенты Zabbix версий до 1.6 больше не поддерживаются. Повышены минимальные требуемые версии баз данных. Минимальная требуемая версия PHP повышена с 8.0.0 до 8.2.0. Символ % добавлен в список `UnsafeUserParameters` для агента Zabbix и агента Zabbix 2.

Обновление с версии	Полные примечания к обновлению	Наиболее важные изменения между версиями
7.2.x	Для: Zabbix 7.4 Zabbix 8.0	Поддержка библиотеки PCRE прекращена.
7.0.x	Для: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Поддержка Oracle DB прекращена.
6.4.x	Для: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия PHP повышена с 7.4.0 до 8.0.0. Асинхронные опросчики для проверок агента, HTTP-агента, SNMP walk[OID] и SNMP get[OID]. Отдельная таблица базы данных для прокси. Изменено расположение файла конфигурации агента Windows по умолчанию. Oracle DB объявлена устаревшей. Старый числовой тип значения (float) удален.
6.2.x	Для: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия MySQL повышена с 8.0.0 до 8.0.30. Для сервера/прокси Zabbix требуется библиотека 'libevent_pthreads'. При первом запуске после обновления прокси Zabbix с SQLite3 автоматически удаляет старую версию базы данных (со всей историей) и создает новую.
6.0.x LTS	Для: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия PHP повышена с 7.2.5 до 7.4.0. Мониторинг сервисов значительно переработан. Во время обновления необходимо создать детерминированные триггеры. Если для MySQL/MariaDB включено двоичное журналирование, требуются привилегии суперпользователя или установка переменной/параметра конфигурации *log_bin_trust_function_creators = 1*. Инструкции по установке переменной см. в разделе Скрипты создания базы данных.
5.4.x	Для: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Повышены минимальные требуемые версии баз данных. Сервер/прокси не запустится при использовании устаревшей базы данных. Записи журнала аудита утрачены из-за изменения структуры базы данных.
5.2.x	Для: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Повышены минимальные требуемые версии баз данных. Составные элементы данных удалены как отдельный тип.
5.0.x LTS	Для: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия PHP повышена с 7.2.0 до 7.2.5. Алгоритм хеширования паролей изменен с MD5 на bcrypt.

Обновление с версии	Полные примечания к обновлению	Наиболее важные изменения между версиями
4.4.x	Для: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Поддержка IBM DB2 прекращена. Минимальная требуемая версия PHP повышена с 5.4.0 до 7.2.0. Повышены минимальные требуемые версии баз данных. Изменено расположение PHP-файлов Zabbix.
4.2.x	Для: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Типы носителей Jabber и Ez Texting удалены.
4.0.x LTS	Для: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Старые прокси больше не могут передавать данные обновленному серверу. Новые агенты больше не смогут работать со старым сервером Zabbix.
3.4.x	Для: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Библиотеки ’libpthread’ и ’zlib’ теперь обязательны. Поддержка протокола в виде обычного текста прекращена, заголовок обязателен. Агенты Zabbix версий до 1.4 больше не поддерживаются. Параметр Server в конфигурации пассивного прокси теперь обязателен.

Обновление с версии	Полные примечания к обновлению	Наиболее важные изменения между версиями
3.2.x	Для: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Поддержка SQLite в качестве внутренней базы данных для сервера/веб-интерфейса Zabbix прекращена. Вместо расширенных регулярных выражений POSIX поддерживаются регулярные выражения, совместимые с Perl (PCRE). Библиотеки 'libpcre' и 'libevent' обязательны для сервера Zabbix. Добавлены проверки кодов завершения для пользовательских параметров, удаленных команд и элементов данных system.run[] без флага 'nowait', а также для скриптов, выполняемых сервером Zabbix. Для поддержки новых функций необходимо обновить Java gateway Zabbix.
3.0.x LTS	Для: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Обновление базы данных может выполняться медленно в зависимости от размера таблицы истории.
2.4.x	Для: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия PHP повышена с 5.3.0 до 5.4.0. Параметр агента LogFile должен быть указан.

Обновление с версии	Полные примечания к обновлению	Наиболее важные изменения между версиями
2.2.x LTS	Для: Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Распределенный мониторинг на основе узлов удален.
2.0.x	Для: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимальная требуемая версия PHP повышена с 5.1.6 до 5.3.0. Для корректной работы сервера требуется регистрозависимая база данных MySQL; для корректной работы сервера Zabbix с базой данных MySQL необходимы кодировка utf8 и сортировка utf8_bin. См. скрипты создания базы данных. Вместо расширения PHP ’mysql’ требуется расширение ’mysqli’.

Обновление из пакетов

Обзор

В этом разделе представлены шаги, необходимые для успешного обновления с использованием официальных пакетов RPM и DEB, поставляемых Zabbix для:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Пакеты Zabbix из репозиториев ОС

Некоторые дистрибутивы ОС (в частности, дистрибутивы на основе Debian) поставляют свои собственные пакеты Zabbix. Эти пакеты **не поддерживаются компанией Zabbix** и могут быть устаревшими, или в них могут отсутствовать последние функции и исправления ошибок. Рекомендуется использовать только официальные пакеты из официального репозитория Zabbix.

Если вы обновляетесь с пакетов, предоставленных дистрибутивами ОС (или они были установлены в какой-то момент), выполните эту процедуру, чтобы перейти на официальные пакеты Zabbix:

1. Удалите старые пакеты.
2. Проверьте на наличие и удалите все оставшиеся файлов из старой инсталляции.
3. Установите официальные пакеты, следуя инструкциям по установке, предоставленным компанией Zabbix.

Не выполняйте прямое обновление, так как такой подход может привести к неработоспособной инсталляции.

Debian/Ubuntu

Обзор

В этом разделе приведены инструкции по обновлению Zabbix **7.4.x** до последней версии Zabbix **8.0.x** с использованием официальных пакетов Zabbix для Debian/Ubuntu.

Warning:
Перед обновлением ознакомьтесь с соответствующими примечаниями по обновлению и убедитесь, что ваша система соответствует требованиям для Zabbix 8.0.

Note:
Рекомендуется использовать две параллельные SSH-сессии во время обновления: одну для выполнения шагов обновления и другую для мониторинга журналов сервера/прокси. Например, во второй сессии выполните `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log`, чтобы в реальном времени просматривать последние записи журнала и возможные ошибки. Это может быть критически важно для продуктивных экземпляров.

Инструкции по обновлению между минорными версиями Zabbix 8.0.x (например, с 8.0.1 до 8.0.3) см. в разделе Обновление между минорными версиями.

Процедура обновления

1. Остановите процессы Zabbix

Остановите Zabbix сервер, чтобы быть уверенным, что в базу данных не будет происходить запись новых данных.

```
systemctl stop zabbix-server
```

Если обновляете Zabbix прокси, агента или агента 2, то остановите и эти компоненты тоже.

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

2. Сделайте резервную копию базы данных Zabbix

Сделайте резервную копию вашей существующей базы данных Zabbix, чтобы подстраховаться от возможных сбоев при обновлении (например, проблем с дисковым пространством, отключения электричества или непредвиденных проблем).

3 Создайте резервную копию файлов конфигурации Zabbix, PHP-файлов и бинарных файлов Zabbix

Создайте резервную копию существующих файлов конфигурации Zabbix, PHP-файлов и бинарных файлов Zabbix.

Для файлов конфигурации выполните:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
#####    Apache:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
#####    Nginx:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

Для PHP-файлов и бинарных файлов Zabbix выполните:

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

4 Обновление пакета конфигурации репозитория

Перед продолжением обновления удалите текущий пакет репозитория Zabbix:

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

Также может потребоваться вручную удалить все старые пакеты Zabbix из рабочего каталога (например, `rm zabbix-release_late` перед загрузкой нового, чтобы менеджер пакетов не использовал устаревшую версию в процессе обновления.

Затем установите последний пакет конфигурации репозитория, чтобы обеспечить совместимость с новейшими пакетами и включить все недавние исправления безопасности или ошибки.

В **Debian 13** выполните:

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian13_all.deb
```

В **Debian 12** выполните:

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

Note:
Для более старых версий Debian замените ссылку выше на правильную из репозитория Zabbix. Однако учтите, что пакеты для этих версий могут не включать все компоненты Zabbix, и для обновления этих компонентов из пакетов стоит рассмотреть обновление ОС. Список включенных компонентов см. в пакетах Zabbix.

В **Ubuntu 26.04** выполните:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu26.04_all.deb
```

В **Ubuntu 24.04** выполните:

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

В **Ubuntu 22.04** выполните:

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

Note:
Для более старых версий Ubuntu замените ссылку выше на правильную из репозитория Zabbix. Однако учтите, что пакеты для этих версий могут не включать все компоненты Zabbix, и для обновления этих компонентов из пакетов стоит рассмотреть обновление ОС. Список включенных компонентов см. в пакетах Zabbix.

Вы можете увидеть запрос, связанный с конфигурацией репозитория Zabbix:

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

Введите `Y` (или `I`), чтобы установить версию файла конфигурации репозитория Zabbix, предоставленную сопровождающим пакета.

Затем обновите сведения о репозитории:

```
apt update
```

5 Обновление компонентов Zabbix

Для обновления компонентов Zabbix выполните:

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

- При использовании PostgreSQL замените `mysql` на `pgsql` в команде.
- При обновлении прокси замените `server` на `proxy` в команде.
- При обновлении агента Zabbix 2 замените `zabbix-agent` на `zabbix-agent2 zabbix-agent2-plugin-*` в команде.

Attention:
Обновление агента Zabbix 2 с помощью команды `apt install zabbix-agent2` может привести к ошибке. Дополнительную информацию см. в разделе *Известные проблемы*.

Может появиться запрос о конфигурации сервера Zabbix (или прокси):

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

Введите вариант, который лучше всего соответствует вашей ситуации. Например, введите `D`, чтобы сравнить текущую и новую конфигурации, а затем решите, хотите ли вы установить версию от сопровождающего пакета (`Y` или `I`).

Затем обновите веб-интерфейс Zabbix и перезапустите веб-сервер.

Для Apache выполните:

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

Для Nginx выполните:

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Для Nginx может появиться запрос о конфигурационном файле `nginx.conf`. Введите вариант, который лучше всего соответствует вашей ситуации:

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

Если после обновления необходимо установить дополнительный пакет Zabbix из того же репозитория, `apt` может выбрать обновление уже установленных пакетов Zabbix во время разрешения зависимостей. Чтобы избежать обновления установленного пакета сервера Zabbix или прокси, временно заблокируйте этот пакет перед установкой, а после установки снимите блокировку:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

6 Проверка параметров конфигурации компонента

Ознакомьтесь с соответствующими примечаниями к обновлению, чтобы проверить, требуются ли какие-либо изменения в параметрах конфигурации.

О новых необязательных параметрах см. страницу Что нового.

Если в качестве веб-сервера используется Nginx и файл конфигурации `/etc/zabbix/nginx.conf` был переустановлен во время обновления, проверьте этот файл и обновите директивы `listen` и `server_name` в соответствии с конфигурацией вашего веб-интерфейса Zabbix.

7. Запустите процессы Zabbix

Запустите обновлённые компоненты Zabbix:

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

8. Очистите cookies и кэш в веб-браузере

После обновления вам, возможно, потребуется очистить cookies веб-браузера и кэш веб-браузера, чтобы веб-интерфейс Zabbix работал должным образом.

Обновление между минорными версиями

Возможно обновление между минорными версиями Zabbix 8.0.x (например, с 8.0.1 до 8.0.3).

Сначала обновите информацию о репозитории:

```
apt update
```

Затем, чтобы обновить все компоненты Zabbix, выполните:

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

- Чтобы обновить только сервер Zabbix, замените `'zabbix*'` на `'zabbix-server*'` в команде.
- Чтобы обновить только прокси Zabbix, замените `'zabbix*'` на `'zabbix-proxy*'` в команде.
- Чтобы обновить только агент Zabbix, замените `'zabbix*'` на `'zabbix-agent*'` в команде.
- Чтобы обновить только агент Zabbix 2, замените `'zabbix*'` на `'zabbix-agent2*'` в команде.

Red Hat Enterprise Linux

Обзор

В этом разделе приведены инструкции по обновлению с Zabbix **7.4.x** до последней версии Zabbix **8.0.x** с использованием официальных пакетов Zabbix для Red Hat Enterprise Linux или его производных - AlmaLinux, CentOS Stream, Oracle Linux и Rocky Linux.

Warning:
Перед обновлением ознакомьтесь с соответствующими примечаниями по обновлению и убедитесь, что ваша система соответствует требованиям для Zabbix 8.0.

Note:
Рекомендуется во время обновления использовать две параллельные SSH-сессии: одну для выполнения шагов обновления, а другую для мониторинга журналов сервера/прокси. Например, во второй сессии выполните `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log`, чтобы в реальном времени просматривать последние записи журнала и возможные ошибки. Это может быть критически важно для рабочих экземпляров.

Инструкции по обновлению между минорными версиями Zabbix 8.0.x (например, с 8.0.1 до 8.0.3) см. в разделе Обновление между минорными версиями.

Процедура обновления

1. Остановите процессы Zabbix

Остановите Zabbix сервер, чтобы быть уверенным, что в базу данных не будет происходить запись новых данных.

```
systemctl stop zabbix-server
```

Если обновляете Zabbix прокси, агента или агента 2, то остановите и эти компоненты тоже.

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

2. Сделайте резервную копию базы данных Zabbix

Сделайте резервную копию вашей существующей базы данных Zabbix, чтобы подстраховаться от возможных сбоев при обновлении (например, проблем с дисковым пространством, отключения электричества или непредвиденных проблем).

3. Сделайте резервную копию файлов конфигурации, PHP и бинарных файлов Zabbix

Сделайте резервную копию существующих файлов конфигурации, файлов PHP и бинарных файлов Zabbix.

Для файлов конфигурации выполните:

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

Для файлов PHP и двоичных файлов Zabbix выполните:

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

4 Обновление пакета конфигурации репозитория

Перед началом обновления установите последнюю версию текущего пакета репозитория, чтобы обеспечить совместимость с новейшими пакетами и получить последние исправления безопасности и ошибок.

В **RHEL 10** выполните:

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

Если вы обновляете прокси Zabbix в RHEL 10 и хотите использовать APM, также установите репозиторий EPEL и включите репозиторий CRB. Оба репозитория предоставляют зависимости для пакета APM прокси Zabbix:

```
dnf -y install epel-release || dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(
crb enable
```

В **RHEL 9** выполните:

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

> **Note:**
> Для более старых версий RHEL или производных от них систем замените приведённую выше ссылку на соответствующую ссылку из репозитория Zabbix. Однако обратите внимание, что пакеты для этих версий могут не включать все компоненты Zabbix. Чтобы обновить такие компоненты из пакетов, рассмотрите возможность обновления ОС. Список включённых компонентов см. в разделе Пакеты Zabbix.

Затем очистите кэш менеджера пакетов `dnf` (включая заголовки, метаданные и файлы пакетов, загруженные во время предыдущих установок или обновлений):

```
dnf clean all
```

При следующей операции `dnf` загрузит свежие метаданные из репозиториев, поскольку старые метаданные были удалены.

См. также: раздел Известные проблемы, посвящённый обновлению пакета конфигурации репозитория в RHEL.

5 Обновление компонентов Zabbix

Для обновления компонентов Zabbix выполните:

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

- При использовании PostgreSQL замените `mysql` на `pgsql` в команде.
- При обновлении прокси замените `server` на `proxy` в команде.
- Если у вас RHEL 10 и вы хотите использовать APM, добавьте `-apm` к имени пакета прокси (например, `zabbix-proxy-mysql-apm`
- При обновлении агента 2 замените `zabbix-agent` на `zabbix-agent2` `zabbix-agent2-plugin-*` в команде.
- Если у вас установлены другие пакеты Zabbix, рассмотрите возможность их обновления, добавив их в команду.

> **Attention:**
> Обновление агента 2 Zabbix с помощью команды `dnf install zabbix-agent2` может привести к ошибке. Дополнительную информацию см. в разделе *Известные проблемы*.

Затем для обновления веб-интерфейса Zabbix с Apache и перезапуска Apache выполните:

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

6 Просмотрите параметры конфигурации компонента

Ознакомьтесь с соответствующими примечаниями к обновлению, чтобы проверить, нужны ли какие-либо изменения в параметрах конфигурации.

С новыми необязательными параметрами можно ознакомиться на странице Что нового.

7. Запустите процессы Zabbix

Запустите обновлённые компоненты Zabbix:

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

8. Очистите cookies и кэш в веб-браузере

После обновления вам, возможно, потребуется очистить cookies веб-браузера и кэш веб-браузера, чтобы веб-интерфейс Zabbix работал должным образом.

Обновление между минорными версиями

Возможно выполнить обновление между минорными версиями Zabbix 8.0.x (например, с 8.0.1 до 8.0.3).

Чтобы обновить все компоненты Zabbix, выполните:

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

- Чтобы обновить только сервер Zabbix, замените `'zabbix-*'` на `'zabbix-server-*'` в команде.
- Чтобы обновить только прокси Zabbix, замените `'zabbix-*'` на `'zabbix-proxy-*'` в команде.
- Чтобы обновить только агент Zabbix, замените `'zabbix-*'` на `'zabbix-agent-*'` в команде.
- Чтобы обновить только агент Zabbix 2, замените `'zabbix-*'` на `'zabbix-agent2-*'` в команде.

Обновление из исходных кодов

Обзор

В этом разделе приведены шаги, необходимые для успешного обновления с Zabbix **7.4**.x до Zabbix **8.0**.x с использованием официальных источников Zabbix.

> **Warning:**
> Перед обновлением обязательно ознакомьтесь с соответствующими **примечаниями к обновлению**!

Также рекомендуется ознакомиться с требованиями для версии 8.0.

> **Note:**
> Во время обновления может быть удобно использовать два параллельных сеанса SSH: в одном выполнять шаги обновления, а в другом отслеживать журналы сервера/прокси. Например, во втором сеансе SSH выполните `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log`, чтобы в режиме реального времени видеть последние записи файла журнала и возможные ошибки. Это может быть особенно важно для рабочих экземпляров.

Процесс обновления сервера

1 Остановите сервер

Остановите сервер Zabbix, чтобы убедиться, что в базу данных не добавляются новые данные.

2 Создайте резервную копию существующей базы данных Zabbix

Это очень важный шаг. Убедитесь, что у вас есть резервная копия базы данных. Она поможет восстановить данные, если процедура обновления завершится с ошибкой (из-за нехватки места на диске, отключения питания или любой другой непредвиденной проблемы).

3 Резервное копирование файлов конфигурации, PHP-файлов и двоичных файлов Zabbix

Создайте резервную копию двоичных файлов Zabbix, файлов конфигурации и каталога PHP-файлов.

4 Установка новых двоичных файлов сервера

Используйте эти инструкции, чтобы скомпилировать сервер Zabbix из исходного кода.

5 Проверьте параметры конфигурации сервера

Обязательно ознакомьтесь с Примечаниями к обновлению, чтобы проверить, требуются ли какие-либо изменения в параметрах конфигурации.

С новыми необязательными параметрами можно ознакомиться на странице Что нового.

6 Запуск новых двоичных файлов Zabbix

Запустите новые двоичные файлы.
Проверьте файлы журналов, чтобы убедиться, что двоичные файлы успешно запустились.

Сервер Zabbix автоматически обновит базу данных.
При запуске сервер Zabbix сообщает текущую (обязательную и необязательную) и требуемую версии базы данных.
Если текущая обязательная версия старше требуемой, сервер Zabbix автоматически выполняет необходимые исправления для обновления базы данных.
Информация о начале обновления базы данных и его ходе (в процентах) записывается в файл журнала сервера Zabbix.
После завершения обновления в файл журнала записывается сообщение "`database upgrade fully completed`".
Если какое-либо исправление для обновления завершится ошибкой, сервер Zabbix не запустится.
Сервер Zabbix также не запустится, если текущая обязательная версия базы данных новее требуемой.
Сервер Zabbix запустится только в том случае, если текущая обязательная версия базы данных соответствует требуемой обязательной версии.

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

Перед запуском сервера:

- Убедитесь, что у пользователя базы данных достаточно прав (создание таблиц, удаление таблиц, создание индексов, удаление индексов)
- Убедитесь, что у вас достаточно свободного места на диске.

7 Установите новый веб-интерфейс Zabbix

Минимально необходимая версия PHP — 8.2.0. При необходимости обновите PHP и следуйте инструкциям по установке.

8 Очистка файлов cookie и кэша веб-браузера

После обновления может потребоваться очистить файлы cookie и кэш веб-браузера, чтобы веб-интерфейс Zabbix работал корректно.

Процесс обновления прокси

1. Останов прокси

Остановите Zabbix прокси.

2. Резервное копирование файлов конфигурации и бинарного файла Zabbix прокси

Выполните резервное копирование бинарного файла и файла конфигурации Zabbix прокси.

3 Установка новых двоичных файлов прокси

Используйте эти инструкции, чтобы скомпилировать прокси Zabbix из исходного кода.

4 Проверьте параметры конфигурации прокси

Обязательно ознакомьтесь с Примечаниями к обновлению, чтобы проверить, требуются ли изменения в параметрах конфигурации.

С новыми необязательными параметрами можно ознакомиться на странице Что нового.

5 Запуск нового прокси Zabbix

Запустите новый прокси Zabbix.
Проверьте файлы журнала, чтобы убедиться, что прокси успешно запущен.

Прокси Zabbix автоматически обновит базу данных.
Обновление базы данных выполняется так же, как при запуске сервера Zabbix.

Процесс обновления агента

> **Attention:**
> Обновление агентов не является обязательным. Обновлять агенты необходимо только в том случае, если это требуется для доступа к новым функциям.

Процедура обновления, описанная в этом разделе, может использоваться для обновления как агента Zabbix, так и агента Zabbix 2.

1. Останов агента

Остановите Zabbix агент.

2. Резервное копирование файлов конфигурации и бинарных файлов Zabbix агента

Выполните резервное копирование бинарного файла и файла конфигурации Zabbix агента.

3 Установите новые двоичные файлы агента

Используйте эти инструкции, чтобы скомпилировать агент Zabbix из исходного кода.

Также можно скачать предварительно скомпилированные агенты Zabbix со страницы загрузки Zabbix.

4 Проверьте параметры конфигурации агента

Обязательно ознакомьтесь с Примечаниями к обновлению, чтобы проверить, требуются ли какие-либо изменения в параметрах конфигурации.

О новых необязательных параметрах см. на странице Что нового.

5 Запуск нового агента Zabbix

Запустите новый агент Zabbix. Проверьте файлы журнала, чтобы убедиться, что агент успешно запущен.

Обновление между минорными версиями

При обновлении между минорными версиями 8.0.x (например, с 8.0.1 до 8.0.3) необходимо выполнить те же действия для сервера/прокси/агента, что и при обновлении между основными версиями. Единственное отличие заключается в том, что при обновлении между минорными версиями изменения в базе данных не вносятся.

Обновление из контейнеров

Обзор

В этом разделе описаны шаги, необходимые для успешного обновления контейнеров Zabbix до версии **8.0**.x.

Отдельные наборы инструкций доступны для:

- Обновление образа компонента Zabbix
- Обновление файла Docker Compose

Warning:
Перед обновлением обязательно ознакомьтесь с соответствующими **замечаниями по обновлению**!

Attention:
Перед началом обновления убедитесь, что у пользователей есть необходимые права доступа к базе данных для выполнения обновления.

 При обновлении с Zabbix 6.0 или более ранней версии во время обновления потребуется создать детерминированные триггеры. Если для MySQL/MariaDB включено двоичное журналирование, для этого требуются привилегии суперпользователя или установка переменной/параметра конфигурации `log_bin_trust_function_creators = 1`. См. Скрипты создания базы данных для инструкций по настройке этой переменной.

 Обратите внимание, что при выполнении из консоли переменная будет установлена только временно и будет сброшена при перезапуске Docker. В этом случае оставьте службу SQL запущенной, остановите только службу `zabbix-server`, выполнив `docker compose down zabbix-server`, а затем `docker compose up -d zabbix-server`.

 В качестве альтернативы вы можете задать эту переменную в файле конфигурации.

В зависимости от размера базы данных обновление базы данных до версии 8.0 может занять много времени.

Обновление образа компонента Zabbix

Эти инструкции позволяют обновить образы Docker, которые используются в качестве основы для контейнеров компонентов Zabbix.

Ниже в качестве примера используется прокси Zabbix.
Замените `zabbix-proxy-sqlite3` на имя контейнера вашего компонента.

1. Проверьте все контейнеры, чтобы определить их имена и текущие версии образов:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest   (...) zabbix-proxy-sqlite3
```

2. Загрузите новую версию образа:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Остановите контейнер компонента Zabbix:

```
docker stop zabbix-proxy-sqlite3
```

4. Удалите контейнер компонента Zabbix:

> **Attention:**
> Перед продолжением убедитесь, что вы записали все флаги конфигурации и точки монтирования томов, использованные в исходной команде `docker run`, так как они понадобятся для воссоздания контейнера.

```
docker rm zabbix-proxy-sqlite3
```

5. Запустите новый контейнер с обновленным образом:

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

> **Note:**
> Дополнительные сведения о запуске контейнеров Zabbix см. в разделе Docker (ручное развертывание).

6. Убедитесь, что обновление прошло успешно, проверив журналы контейнера:

```
docker logs -f zabbix-proxy-sqlite3
```

Обновление файла Docker Compose

Эти инструкции предназначены для обновления файлов Docker Compose, используемых для развертывания компонентов Zabbix. Следуйте этим инструкциям, если вы установили Zabbix с помощью Docker Compose.

1. Проверьте все контейнеры, чтобы определить их имена и текущие версии образов:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                               (...) zabbix-docker-mysql-server-1
```

2. Перейдите в репозиторий Zabbix Docker, который вы ранее клонировали, получите последние обновления и переключитесь на ветку 8.0:

> **Attention:**
> Если вы изменили `.env`, `compose.yaml` или другие файлы в локальном репозитории, команда `git pull` может перезаписать их. Создайте резервную копию всех локальных изменений перед выполнением этого шага.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Разверните компоненты Zabbix с использованием обновленного файла Docker Compose:

```
####   MySQL              :
docker compose -f ./compose.yaml up -d

####   PostgreSQL               :
docker compose -f ./compose_pgsql.yaml up -d
```

> **Note:**
> После того как все контейнеры будут пересозданы и запущены, возобновление сбора данных может занять 1-3 минуты. Дополнительную информацию о запуске контейнеров Zabbix см. в разделе Docker Compose.

4. Убедитесь, что обновление прошло успешно, проверив журналы контейнера (замените `zabbix-docker-zabbix-server-1` на имя контейнера вашего компонента):

```
docker logs -f zabbix-docker-zabbix-server-1
```

Проблемы при обновлении

Экранирование и связанные с ним аспекты Превышено ограничение на количество символов для параметров функции после обновления

В Zabbix 7.0 добавлено правильное экранирование обратных косых черт в строковых параметрах функций истории.

Поскольку при обновлении с версий Zabbix до 7.0 добавляются дополнительные обратные косые черты, это приводит к увеличению длины параметров, что может привести к поломке функций триггера, если длина параметра превышает максимальный размер данных в 255 символов.

Чтобы проиллюстрировать, какие проблемы могут возникнуть при обновлении базы данных, каково их влияние, как их распознать и исправить, давайте рассмотрим пример настройки с тремя элементами данных и двумя триггерами.

Настройка

1. Версия Zabbix 6.0.
2. Файл `/tmp/ONE`, содержащий 114 символов обратной косой черты и символ «a» в конце:

```
\\\
```

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

С вычисляемыми элементами данных проблем нет, но их также можно обновить для согласованности:

Calc_228:

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

Calc_232:

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

Известные проблемы

См. также: Проблемы компиляции.

Обновление

Настройки режима SQL для успешного обновления

Параметр `sql_mode` в MySQL/MariaDB должен быть выставлен в режим «STRICT_TRANS_TABLES». Если он отсутствует, обновление базы данных Zabbix завершится неудачей (смотрите также ZBX-19435 [en]).

Обновление с MariaDB 10.2.1 и старше

Обновление Zabbix может завершится с ошибкой, если таблицы базы данных были созданы с MariaDB версии 10.2.1 и старше, так как в этих версиях по умолчанию используется компактный формат строк. Такая проблема исправляется изменением формата строк на динамический формат (смотрите также ZBX-17690 [en]).

Кодировка и сортировка базы данных MySQL

В установках Zabbix, использующих базу данных MySQL, созданную с кодировкой utf8mb3 и сортировкой utf8mb3_bin, могут возникать проблемы. Например, известная проблема MySQL может приводить к ошибкам `Row size too large (> 8126)` во время обновления схемы базы данных или выполнения других операций DDL.
 Настоятельно рекомендуется преобразовать базы данных с кодировкой utf8mb3 и сортировкой utf8mb3_bin в кодировку utf8mb4 и сортировку utf8mb4_bin, как описано в разделе Восстановление кодировки и сортировки базы данных Zabbix.

Утечка памяти и дескрипторов файлов с libcurl 8.21.0

Известная проблема в libcurl 8.21.0 может вызывать утечки памяти и дескрипторов файлов в функциях Zabbix, использующих libcurl при обработке соединений, защищенных TLS.
 Временным обходным решением для этой проблемы является понижение версии libcurl до 8.20.0, пока не станет доступен выпуск с исправлением.

Шаблоны

Совместимость шаблонов в двухстековых средах (IPv4/IPv6)

В двухстековых средах (системах, настроенных для поддержки как IPv4, так и IPv6) имя узла `localhost` обычно разрешается как в IPv4-, так и в IPv6-адреса. Из-за распространенного приоритета IPv6 над IPv4 во многих операционных системах и DNS-резолверах шаблоны Zabbix могут работать некорректно, если отслеживаемый сервис настроен на прослушивание только IPv4.

Сервисы, не настроенные на прослушивание IPv6-адресов, могут оказаться недоступными, что приведет к сбоям мониторинга. Пользователи могут правильно настроить доступ для IPv4, но все равно столкнуться с проблемами подключения из-за поведения по умолчанию, при котором приоритет отдается IPv6.

Один из способов обойти эту проблему — убедиться, что сервисы (Nginx, Apache, PostgreSQL и т. д.) настроены на прослушивание как IPv4-, так и IPv6-адресов, а серверу/агенту Zabbix разрешен доступ по IPv6. Кроме того, в шаблонах и конфигурациях Zabbix следует явно использовать `localhost` вместо `127.0.0.1`, чтобы обеспечить совместимость как с IPv4, так и с IPv6.

Например, при мониторинге PostgreSQL с помощью шаблона PostgreSQL by Zabbix agent 2 может потребоваться отредактировать файл `pg_hba.conf`, чтобы разрешить подключения для пользователя `zbx_monitor`. Если в двухстековой среде приоритет отдается IPv6 (система разрешает localhost в `::1`) и вы указываете `localhost`, но добавляете только запись IPv4 (`127.0.0.1/32`), подключение завершится ошибкой, поскольку не будет соответствующей записи IPv6.

Следующий пример файла `pg_hba.conf` гарантирует, что пользователь `zbx_monitor` сможет подключаться к любой базе данных с локальной машины, используя как IPv4-, так и IPv6-адреса с разными методами аутентификации:

```
### TYPE      DATABASE      USER             ADDRESS          METHOD
  host      all           zbx_monitor      localhost        trust
```

```
host    all             zbx_monitor     127.0.0.1/32    md5
host    all             zbx_monitor     ::1/128         scram-sha-256
```

При необходимости вы также можете использовать IPv4-адрес (`127.0.0.1`) напрямую при настройке макроса шаблона PostgreSQL by Zabbix agent 2 для строки подключения.

Случайная установка пакетов Zabbix из EPEL

Если репозиторий EPEL установлен и активирован, установка пакетов Zabbix может привести к загрузке версий EPEL вместо официальных пакетов Zabbix. Чтобы решить эту проблему:

1. Удалите все пакеты Zabbix, установленные из EPEL:

```
dnf remove zabbix-server-mysql
```

2. Исключите пакеты Zabbix из EPEL, добавив следующую строку в файл `/etc/yum.repos.d/epel.repo`:

```
[epel]
...
excludepkgs=zabbix*
```

3. Переустановите официальный пакет сервера Zabbix:

```
dnf install zabbix-server-mysql
```

Во время установки официальные пакеты Zabbix включают слово `release` в строку версии (например, `7.0.0-release1.el8`), что отличает их от пакетов EPEL.

Пакеты Zabbix для RHEL в средах Red Hat UBI

При установке Zabbix из пакетов Red Hat Enterprise Linux в средах Red Hat Universal Base Image убедитесь, что есть доступ к необходимым репозиториям и зависимостям. Пакеты Zabbix зависят от библиотек `libOpenIPMI.so` и `libOpenIPMIposix.so`, которые не предоставляются ни одним пакетом в репозиториях менеджера пакетов по умолчанию, включенных в системах UBI, и это приведет к ошибкам установки.

Библиотеки `libOpenIPMI.so` и `libOpenIPMIposix.so` доступны в пакете `OpenIPMI-libs`, который предоставляется репозиторием `redhat-#-for-<arch>-appstream-rpms`. Доступ к этому репозиторию определяется подписками, которые в случае сред UBI передаются путем монтирования каталогов конфигурации репозиториев и секретов хоста RHEL в пространство имен файловой системы контейнера.

Дополнительную информацию см. в ZBX-24291.

Истёкший ключ подписи для пакетов RHEL

При обновлении Zabbix на Red Hat Enterprise Linux или его производных вы можете столкнуться с проблемой истечения срока действия ключа подписи пакетов в репозитории Zabbix. Когда срок действия ключа подписи истекает, попытки проверить подписи пакетов приводят к ошибке, указывающей на то, что сертификат или ключ больше недействителен. Например:

```
error: Verifying a signature using certificate D9AA84C2B617479C6E4FCF4D19F2475308EFA7DD (Zabbix LLC (Jul 2
  1. Certificate 19F2475308EFA7DD invalid: certificate is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
  2. Key 19F2475308EFA7DD invalid: key is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
```

Чтобы устранить такие проблемы, вручную переустановите последний пакет `zabbix-release` для вашего конкретного варианта RHEL (замените ссылку ниже на правильную из репозитория Zabbix).

Например, в **RHEL 10** выполните:

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

Затем обновите информацию о репозиториях:

```
dnf update
```

Дополнительную информацию см. в ZBX-24761.

TLS-подключение к базе данных с MariaDB

TLS-подключение к базе данных не поддерживается с опцией `verify_ca` для параметра DBTLSConnect, если используется MariaDB.

Возможные взаимные блокировки с MySQL/MariaDB

При работе под высокой нагрузкой и использовании более одного процесса LLD worker возможно возникновение взаимных блокировок (deadlocks), вызванных ошибкой InnoDB, относящейся к стратегии блокировок строк (см. upstream bug [en]). Ошибка была исправлена в MySQL, начиная с версии 8.0.29, но не в MariaDB. Для более подробной информации, обратитесь к ZBX-21506 [en].

Глобальная корреляция событий

События могут не коррелироваться должным образом, если промежуток времени между первым и вторым событием очень мал, то есть полсекунды и менее.

NetBSD 8.0 и новее

Различные процессы Zabbix могут случайным образом аварийно завершать работу при запуске на NetBSD версий 8.X и 9.X. Такое поведение связано со слишком малым размером стека по умолчанию (4МБ), который необходимо увеличить, выполнив:

```
ulimit -s 10240
```

Для получения более подробной информации, пожалуйста, обратитесь к соответствующему отчёту о проблеме: ZBX-18275 [en].

Ограничения регулярных выражений в Zabbix агенте 2

Агент Zabbix 2 не поддерживает просмотр вперед и назад в регулярных выражениях из-за ограничений стандартной библиотеки регулярных выражений Go.

IPMI проверки

IPMI проверки не будут работать со стандартным пакетом библиотеки OpenIPMI на Debian до версии 9 (stretch) и Ubuntu до 16.04 (xenial). Чтобы исправить проблему, пересоберите OpenIPMI библиотеку с включённым OpenSSL, как обсуждалось в ZBX-6139 [en].

IPMI — недоверенные узлы сети могут привести к сбою OpenIPMI

В библиотеке OpenIPMI, используемой Zabbix для опроса данных IPMI, существует ошибка, которую можно вызвать специально сформированными ответами от недоверенного устройства.

Недоверенное устройство IPMI может отправить специально сформированные данные, из-за которых библиотека OpenIPMI завершит работу с ошибкой. Это, в свою очередь, может привести к завершению процесса сервера Zabbix, выполняющего опрос IPMI.

SSH проверки

- Некоторые дистрибутивы Linux, такие как Debian и Ubuntu, не поддерживают зашифрованные закрытые (private) ключи (с парольной фразой), если библиотека libssh2 установлена из пакетов. Пожалуйста, обратитесь к ZBX-4850 [en] для получения более подробных сведений.
- При использовании libssh 0.9.x на некоторых дистрибутивах Linux с OpenSSH 8 проверки SSH могут иногда сообщать: «Cannot read data from SSH server». Такое поведение вызвано проблемой [en](более подробный отчёт [en]) libssh. Ожидается, что эта ошибка исправлена в стабильном выпуске libssh 0.9.5. Смотрите также ZBX-17756 [en] для получения подробной информации.
- Использование вертикальной черты "|" в SSH скрипте может привести к ошибке «Cannot read data from SSH server». В этом случае рекомендуется обновить версию библиотеки libssh. Смотрите также ZBX-21337 [en] для получения подробной информации.

Проверки ODBC

- Драйвер MySQL unixODBC не следует использовать с сервером Zabbix или прокси Zabbix, собранными с библиотекой MariaDB connector, и наоборот; если возможно, также лучше избегать использования того же connector, что и драйвер, из-за ошибки upstream. Рекомендуемая настройка:

  PostgreSQL, SQLite или Oracle connector → драйвер MariaDB или MySQL unixODBC
MariaDB connector → драйвер MariaDB unixODBC
MySQL connector → драйвер MySQL unixODBC

  См. ZBX-7665 для получения дополнительной информации и доступных обходных решений.

- XML-данные, запрашиваемые из Microsoft SQL Server, могут обрезаться различными способами в системах Linux и UNIX.

- Было замечено, что использование проверок ODBC для мониторинга баз данных Oracle с различными версиями Oracle Instant Client для Linux приводит к аварийному завершению работы сервера Zabbix.
 См. также: ZBX-18402, ZBX-20803.
- При использовании драйвера FreeTDS UnixODBC необходимо добавить оператор 'SET NOCOUNT ON' в начало SQL-запроса (например, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). В противном случае элемент данных мониторинга базы данных в Zabbix не сможет получить информацию и вернет ошибку "SQL query returned empty result".
 См. ZBX-19917 для получения дополнительной информации.

Некорректный параметр метода запроса в элементах данных

Параметр метода запроса, используемый только в HTTP проверках, может быть некорректным образом задан значением '1', значением не по умолчанию для всех элементов данных, в результате обновления с версии Zabbix pre-4.0. Для получения более подробных сведений по устранению данной ситуации смотрите ZBX-19308 [en].

Веб-мониторинг и HTTP агент

Zabbix сервер имеет утечку памяти на некоторых дистрибутивах Linux по причине ошибки в библиотеке [en] при включённой опции «Проверка SSL узла» в веб-сценариях или HTTP агенте. Пожалуйста, обратитесь к ZBX-10486 [en] для получения более подробной информации и возможных вариантов решения.

Простые проверки

Имеется проблема в утилите **fping** версий до v3.10, которые некорректно обрабатывают дубликаты ответных echo пакетов. Такое поведение может вызывать непредвиденные результаты в элементах данных `icmpping`, `icmppingloss`, `icmppingsec`. Рекомендуется использовать последнюю версию **fping**. Пожалуйста, обратитесь к ZBX-11726 [en] для получения более подробной информации.

Ошибки при выполнении fping в контейнерах без root-доступа

Когда контейнеры работают в режиме без root-доступа или в среде со специальными ограничениями, вы можете столкнуться с ошибками, связанными с работой fping при выполнении проверок ICMP, такими как: «`fping: Operation not permitted`», — или потерей всех пакетов ко всем ресурсам.

Для исправления этой проблемы добавьте `--cap-add=net_raw` к командам «docker run» или «podman run».

В дополнение к этому, выполнение fping в средах без root-доступа может потребовать модификации sysctl, то есть:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

где «1995» — это GID zabbix. Для дополнительной информации обратитесь к ZBX-22833 [en].

SNMP проверки

При использовании операционной системы OpenBSD, проблема «использование памяти после её освобождения» в библиотеке Net-SNMP вплоть до версии 5.7.3 может привести к аварийной остановке сервера Zabbix, если в файле конфигурации сервера Zabbix указан параметр SourceIP. Как временное решение, пожалуйста, не задавайте параметр SourceIP. Эта проблема также применима и к Linux, но она не приводит к остановке работы сервера Zabbix. К пакету net-snmp применен локальный патч на OpenBSD и будет выпущен с версией OpenBSD 6.3.

Резкие скачки в SNMP данных

Наблюдались резкие скачки в SNMP данных, которые могут быть связаны с определёнными физическими факторами, такими как скачки напряжения в сети. Обратитесь к ZBX-14318 [en] для получения более подробной информации.

SNMP трапы

Пакет «net-snmp-perl», требуемый для SNMP трапов, удалён в RHEL 8.0-8.2; добавлен повторно в RHEL 8.3.

Таким образом, если вы используете RHEL 8.0-8.2, лучшим решением будет обновление до RHEL 8.3.

Пожалуйста, также просмотрите ZBX-17192 [en] для получения более подробной информации.

Сбой в процессе alerter в RHEL 7

В RHEL 7 обнаружены случаи сбоя в процессах alerter сервера Zabbix. Пожалуйста, обратитесь к ZBX-10461 [en] для получения более подробной информации.

Обновление Zabbix агента 2 (6.0.5 или более старого)

При обновлении Zabbix агента 2 (версии 6.0.5 или старше) из пакетов может возникнуть ошибка конфликта файлов, связанных с плагином. Чтобы исправить ошибку, сделайте резервную копию конфигурации агента 2 (если необходимо), деинсталлируйте агент 2 и установите заново.

На системах на базе RHEL выполните:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

На системах на базе Debian выполните:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Для более подробной информации обратитесь к ZBX-23250 [en].

Переключение локалей веб-интерфейса

Было замечено, что локали веб-интерфейса могут переключаться без видимой логики: некоторые страницы (или их части) отображаются на одном языке, а другие страницы (или их части) - на другом.

Обычно проблема может возникать, если имеется несколько пользователей, одни из которых используют одну локаль, а другие - другую.

Известный способ устранения этой проблемы - отключить многопоточность в PHP и Apache.

Проблема связана с тем, как работает установка локали в PHP: информация о локали хранится для процесса, а не для потока. Поэтому в многопоточной среде, когда одним процессом Apache запускается несколько проектов, локаль может быть изменена в другом потоке, что влияет на обработку данных в потоке Zabbix.

Дополнительную информацию см. в связанных отчетах о проблемах:

- ZBX-10911 (Проблема с переключением локалей веб-интерфейса)
- ZBX-16297 (Проблема с обработкой чисел на графиках при использовании функции `bcdiv` из функций BC Math)

Графики

Проблемы с графиками (классическими)

Если у вас возникли проблемы с классическими графиками, рекомендуется обновить библиотеку GD (libgd) до версии 2.3.3-13 или выше, а PHP — до версии 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 или выше.

Летнее время

Переходы на летнее время (DST) приводят к нарушению отображения подписей к оси X (дублирование даты, отсутствие даты и т.п.).

Агрегирование в виде суммы

При использовании агрегирования в виде суммы в графике за период менее одного часа, графики отображают некорректные (размноженные) значения, когда данные поступают из динамики изменений.

Наложение текста

Для некоторых языков веб-интерфейса (например, японского), местные шрифты могут приводить к наложению текста в легенде графика. Чтобы избежать этого, используйте версию 2.3.0 (или более позднюю) расширения PHP GD.

Мониторинг файлов журналов

Элементы данных `log[]` и `logrt[]` многократно перечитывают файл журнала с самого начала, если файловая система заполнена на 100% и файл журнала продолжает заполняться (обратитесь к ZBX-10884 [en] для получения более подробной информации).

Медленные запросы MySQL

Zabbix сервер генерирует медленные запросы `SELECT` в случае несуществующих значений у элементов данных. Это обусловлено известной проблемой [en] в MySQL версий 5.6/5.7 (для её обсуждения обратитесь к ZBX-10652 [en]), а также, в некоторых случаях, может происходить и в более поздних версиях MySQL. Чтобы обойти эту проблему, отключите параметр `index_condition_pushdown` [en] или `prefer_ordering_index` [en] оптимизатора в MySQL. Обратите внимание, однако, что такой обходной путь может исправить не все проблемы, относящиеся к медленным запросам.

Постоянные настройки фильтра из ссылок

При открытии ссылки на страницу веб-интерфейса Zabbix, содержащую настройки фильтра, включая выбор времени, фильтр автоматически сохраняется в базе данных для пользователя, заменяя ранее сохраненные настройки фильтра и/или выбора времени для этой страницы. Эти настройки остаются активными, пока пользователь не обновит или не сбросит их вручную.

Проблема с IPv6 адресами в SNMPv3 трапах

По причине проблемы в net-snmp, в SNMP трапах IPv6 адреса могут некорректно отображаться при использовании SNMPv3. Для получения более подробной информации и возможных вариантах решения смотрите ZBX-14541 [en].

Урезание длинных IP адресов IPv6 в информации о неуспешном входе в систему

Сообщение об ошибочном входе в систему отобразит только первые 39 символов сохранённого IP адреса, так как это ограничение длины поля базы данных. Это означает, что IP адреса IPv6 длиннее 39 символов будут отображаться не полностью.

Проверки агента Zabbix в Windows

Несуществующие записи DNS в параметре `Server` файла конфигурации агента Zabbix (`zabbix_agentd.conf`) могут увеличивать время отклика агента Zabbix в Windows. Это происходит потому, что демон кэширования DNS в Windows не кэширует отрицательные ответы для IPv4-адресов. Однако для IPv6-адресов отрицательные ответы кэшируются, поэтому возможным обходным решением является отключение IPv4 на узле сети.

Экспорт/импорт YAML

Существуют некоторые известные проблемы с YAML экспортом/импортом:

- Сообщения об ошибках не подлежат переводу;
- Допустимый JSON с расширением файла .yaml иногда не удается импортировать;
- Не заключенные в кавычки человекочитаемые даты автоматически преобразуются в Unix-метки времени.

Мастер настройки на SUSE с NGINX и php-fpm

Мастер настройки веб-интерфейса не может сохранить файл конфигурации на SUSE с NGINX + php-fpm. Это вызвано параметром в unit-файле `/usr/lib/systemd/system/php-fpm.service`, который не позволяет Zabbix записывать в `/etc`. (появилось в PHP 7.4).

Доступны два варианта обходного решения:

- Установите для параметра ProtectSystem значение 'true' вместо 'full' в unit-файле php-fpm systemd.
- Сохраните файл `/etc/zabbix/web/zabbix.conf.php` вручную.

Перенаправление заголовка Authorization

В некоторых случаях Apache или NGINX могут препятствовать тому, чтобы заголовок Authorization в API-запросах доходил до Zabbix. Это может вызывать проблемы с аутентификацией при использовании Zabbix API или служб единого входа (SSO), таких как SAML с Okta.

Чтобы устранить это, обновите конфигурацию вашего веб-сервера.

Для **Apache**, если вы используете его как обратный прокси (не CGI-конфигурация), добавьте следующую директиву в `/etc/httpd/conf/httpd.conf` (в системах на базе RHEL) или `/etc/apache2/apache2.conf` (в Debian/Ubuntu):

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Если Apache напрямую выполняет скрипты для обработки запросов (например, с помощью mod_cgi), вместо этого добавьте следующую директиву:

```
CGIPassAuth On
```

В отличие от этого, **NGINX** обрабатывает заголовок Authorization автоматически. Однако, если NGINX работает как обратный прокси, вы можете явно перенаправить заголовок Authorization, добавив следующие директивы в `/etc/nginx/nginx.conf` (для расположения веб-интерфейса Zabbix):

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

После обновления конфигурации перезапустите ваш веб-сервер.

Для получения дополнительных сведений см.:

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- директивы SetEnvIfNoCase и CGIPassAuth
- NGINX Reverse Proxy

Chromium для веб-сервиса Zabbix в Ubuntu 20

Хотя в большинстве случаев веб-сервис Zabbix может работать с Chromium, в Ubuntu 20.04 использование Chromium приводит к следующей ошибке:

```
Cannot fetch data: chrome failed to start:cmd_run.go:994:
WARNING: cannot create user data directory: cannot create
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

Эта ошибка возникает из-за того, что в качестве домашней директории пользователя «zabbix» используется `/var/lib/zabbix`.

Специальные коды ошибок MySQL

Когда Zabbix обнаруживает, что внутренняя база данных недоступна, он отправляет уведомление и продолжает попытки подключения. Для определённых движков баз данных распознаются определённые коды ошибок. В MySQL эти распознаваемые коды ошибок включают:

- CR_CONN_HOST_ERROR
- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

Кроме того, если Zabbix используется с MySQL инсталляцией в Azure, в журналах Zabbix может появиться расплывчатое сообщение об ошибке *[9002] Some errors occurred*. Это сообщение отправляется Zabbix серверу или прокси со стороны базы данных. Чтобы определить причину ошибки, обратитесь к журналам Azure.

Недопустимые регулярные выражения после перехода на PCRE2

В Zabbix 6.0 была добавлена поддержка PCRE2. Хотя PCRE по-прежнему поддерживается, установочные пакеты Zabbix для RHEL 7 и новее, SLES (все версии), Debian 9 и новее, Ubuntu 16.04 и новее были обновлены для использования PCRE2. Несмотря на множество преимуществ, переход на PCRE2 может привести к тому, что некоторые существующие шаблоны регулярных выражений PCRE станут недопустимыми или будут работать иначе. В частности, это касается шаблона *[\w-\.]*. Чтобы снова сделать это регулярное выражение допустимым без изменения семантики, измените выражение на *[-\w\.]* . Это происходит из-за того, что PCRE2 рассматривает знак дефиса как разделитель, создавая диапазон внутри символьного класса.

Ошибка виджета Geomap

Карты в виджете Geomap могут загружаться некорректно, если вы обновились с более старой версии Zabbix с NGINX и не переключились на новый файл конфигурации NGINX во время обновления.

Чтобы исправить проблему, вы можете удалить старый файл конфигурации, использовать файл конфигурации из пакета текущей версии и перенастроить его, как описано в инструкции по загрузке в разделе *e. Configure PHP for Zabbix frontend*.

В качестве альтернативы вы можете вручную отредактировать существующий файл конфигурации NGINX (обычно `/etc/zabbix/nginx.conf`). Для этого откройте файл и найдите следующий блок:

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

Затем замените этот блок на:

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

Предобработка — глобальные переменные небезопасны

JavaScript в предобработке выполняется для каждого запроса, но присваивания не объявленным идентификаторам (например, `secret = value`) создают неявные глобальные переменные, которые могут сохраняться дольше текущего выполнения. Хранение конфиденциальных данных (токенов, паролей и т. д.) в неявных глобальных переменных

повышает риск случайного раскрытия или повторного использования при последующих запусках предобработки либо другими интеграциями, выполняющимися в той же среде.

Не полагайтесь на неявные глобальные переменные. Всегда объявляйте переменные с помощью `var` или `const` и не привязывайте секреты к глобальным объектам (например, `globalThis` или `window`). В рамках предобработки не предусмотрен поддерживаемый способ переопределить встроенные глобальные объекты.

Безопасный пример:

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

Группы процессоров в Windows

В документации Microsoft указано, что системы с менее чем 64 логическими процессорами всегда имеют одну группу процессоров, Group 0. Однако пользователи Zabbix сообщили о редкой ошибке ZBX-20260, когда на системах с 64 или менее логическими процессорами присутствуют две группы процессоров. В результате счетчики производительности "\Processor(n)" были доступны только для одной из двух групп процессоров. Фактическая первопричина этой ошибки неизвестна. Однако похожий случай был описан на stackoverflow.com, и там первопричина заключалась во взаимодействии между BIOS и Windows.

Ограничения фильтрации с utf8mb4-сопоставлениями

Фильтры (например, в *Сбор данных* > *Обслуживание*) могут работать некорректно при применении к сущностям, содержащим определенные символы Unicode (например, 𝓀, 𝓁). Эта проблема возникает из-за того, как сопоставление utf8mb4_bin по умолчанию для баз данных MySQL или MariaDB обрабатывает сортировку и сравнение символов Unicode.

Чтобы обойти это ограничение, пользователи могут изменить сопоставление столбцов базы данных на альтернативы, такие как utf8mb4_0900_bin, utf8mb4_0900_ai_ci или utf8mb4_unicode_520_ci. Однако следует учитывать, что изменение сопоставления может привести к неожиданному поведению при обработке пустых пробелов, а также при сортировке и фильтрации других символов.

Дополнительную информацию об изменении сопоставлений см. в документации MySQL или документации MariaDB. Подробности о различиях между сопоставлениями см. в разделе Unicode Character Sets в документации MySQL.

Профилирование чрезмерного использования памяти с помощью tcmalloc

Если вы подозреваете, что ваша установка Zabbix использует слишком много памяти, вы можете воспользоваться функцией профилирования памяти tcmalloc, чтобы исследовать потребление памяти сервером/прокси Zabbix.

1. При установке Zabbix из исходных кодов задайте дополнительные флаги:

```
export CFLAGS="-std=gnu99 -g -O0"
```

Флаг `-std=gnu99` требуется для сборки сервера Zabbix, прокси Zabbix или агента Zabbix. Флаг `-g` добавляет дополнительную отладочную информацию, а `-O0` отключает оптимизации, которые могут мешать профилированию tcmalloc.

2. Перед запуском сервера Zabbix задайте следующие переменные окружения. Эти переменные сообщают tcmalloc, как отслеживать и сообщать об использовании памяти:

```
LD_PRELOAD="/usr/lib/aarch64-linux-gnu/libtcmalloc.so" \
HEAPPROFILE=./heap_profile \
HEAP_PROFILE_ALLOCATION_INTERVAL=0 \
HEAP_PROFILE_INUSE_INTERVAL=4294967296 \
HEAPPROFILESIGNAL=5 \
MALLOCSTATS=1 \
./sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

3. Создайте дамп профиля, отправив сигнал 5 целевому процессу. Замените 1234 на фактический идентификатор процесса (PID):

```
kill -5 1234
```

4. Выведите сгенерированный профиль:

```
pprof-symbolize -text ./sbin/zabbix_server ./heap_profile.0001.heap

Using local file ./sbin/zabbix_server.
Using local file ./heap_profile.0001.heap.
Total: 1078.1 MB
```

```
  1076.8  99.9%  99.9%   1076.8  99.9% zbx_malloc2
     1.0   0.1% 100.0%      1.0   0.1% __GI___strdup
     0.2   0.0% 100.0%      0.2   0.0% CRYPTO_zalloc@@OPENSSL_3.0.0
     0.1   0.0% 100.0%      0.1   0.0% OPENSSL_LH_insert@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% zbx_realloc2
     0.0   0.0% 100.0%      0.1   0.0% PKCS7_decrypt@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% find_best_tree_node
     0.0   0.0% 100.0%      0.0   0.0% CRYPTO_strndup@@OPENSSL_3.0.0
     ...
     0.0   0.0% 100.0%      0.0   0.0% preprocessing_flush_value
     0.0   0.0% 100.0%   1074.0  99.6% preprocessor_add_request
```

В этом примере zbx_malloc2 отвечает почти за все выделения памяти.

См. также:

- ZBX-25050 и ZBX-25584 — связанные отчеты о проблемах.
- GCC Option Summary — о параметрах компиляции (`-std=gnu99`, `-g`, `-O0` и т. д.).
- Gperftools Heap Profiler — документация по переменным окружения для профилирования tcmalloc.

MySQL 8.0 Group Replication в режиме multi-primary

При использовании MySQL 8.0 Group Replication в режиме multi-primary во время фиксации транзакций может возникать ошибка, похожая на следующую:

```
1531697:20250128:064734.697 query [txnlev:1] [update alerts set status=1,retries=0,error='' where alertid=
1531697:20250128:064734.713 query [txnlev:1] [commit;]
1531697:20250128:064734.753 [Z3005] query failed: [3101] Plugin instructed the server to rollback the curr
```

Похоже, эта ошибка вызывается проблемами с операциями отката, связанными с ограничениями внешнего ключа.

См. также:

- ZBX-26060 — связанный отчет о проблеме.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" — соответствующая проблема в upstream.

Вход в систему и systemd

Мы рекомендуем создать пользователя *zabbix* в качестве системного пользователя, то есть без возможности входа в систему. Некоторые пользователи игнорируют эту рекомендацию и используют ту же учетную запись для входа в систему (например, через SSH) на узел сети, на котором запущен Zabbix. Это может привести к аварийному завершению работы демона Zabbix при выходе из системы. В этом случае в журнале сервера Zabbix появится примерно следующее:

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

а в журнале агента Zabbix:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Это происходит из-за настройки systemd по умолчанию `RemoveIPC=yes`, заданной в `/etc/systemd/logind.conf`. При выходе из системы ранее созданные Zabbix семафоры удаляются, что приводит к аварийному завершению работы.

Цитата из документации systemd:

```
RemoveIPC=

     ,            IPC System V   POSIX,               ,
                      .                 .             ,
                         IPC                                  .
            ,                        System V,
          POSIX.           ,         IPC         root
                               .               — "yes".
```

Существует 2 решения этой проблемы:

1. (рекомендуется) Не используйте учетную запись *zabbix* ни для чего, кроме процессов Zabbix; создайте отдельную учетную запись для других задач.
2. (не рекомендуется) Установите `RemoveIPC=no` в `/etc/systemd/logind.conf` и перезагрузите систему. Обратите внимание, что `RemoveIPC` — это общесистемный параметр, и его изменение повлияет на всю систему.

Требования

Аппаратное обеспечение

Память

Zabbix требуется как физическая память, так и память на диске. Очевидно, что объём необходимой памяти на диске зависит от количества наблюдаемых узлов сети и наблюдаемых параметров. Если вы планируете достаточно долгосрочное хранение истории наблюдаемых параметров, то потребуется по крайней мере несколько гигабайт для хранения данных истории в базе данных. Каждый из процессов демона Zabbix требует несколько подключений к серверу базы данных. Объём памяти, требуемый на каждое из подключений к базе данных, зависит от настроек базы данных.

Note:
Чем больше оперативной памяти вам доступно, тем быстрее работает база данных (а следовательно, и Zabbix).

CPU

Zabbix и особенно база данных могут потребовать значительных ресурсов процессора в зависимости от количества наблюдаемых параметров и выбранного типа базы данных.

Другое оборудование

Для использования встроенных в Zabbix SMS уведомлений потребуется последовательный порт передачи данных и GSM модем. Конвертер USB-to-serial также будет работать.

Примеры конфигурации оборудования

В таблице приведены примеры конфигурации оборудования для платформы **Linux/BSD/Unix**.

Это примеры размеров и конфигураций оборудования, с которых можно начать. Каждая установка Zabbix уникальна. Обязательно протестируйте производительность системы Zabbix в промежуточной или среде разработки, чтобы полностью понять свои требования до развертывания установки Zabbix в рабочей среде.

Размер установки	Контролируемые метрики[1]	Ядра CPU/vCPU	Память (ГиБ)	База данных	Amazon EC2[2]
Малый	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Средний	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Большой	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Очень большой	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] Для определения размера 1 контролируемая метрика приблизительно представляется формулой: 1 метрика = 1 элемент данных + 1 триггер + 1 график. Это лишь оценочное правило, а не формула для расчета количества контролируемых метрик.
 [2] Пример с инстансами Amazon EC2 общего назначения, использующими архитектуру ARM64 или x86_64. При оценке и тестировании установки Zabbix перед развертыванием в рабочей среде следует выбрать подходящий тип инстанса, например оптимизированный для вычислений, памяти или хранилища.

Note:
Фактическая конфигурация в значительной степени зависит от количества активных элементов данных и частоты их обновления (подробности см. в разделе Размер базы данных на этой странице). Для крупных установок настоятельно рекомендуется запускать базу данных на отдельном сервере.

Поддерживаемые платформы

В связи с требованиями безопасности и критически важным характером работы системы мониторинга, единственной операционной системой, которая может обеспечить необходимую производительность, отказоустойчивость и гибкость является операционная система UNIX. Zabbix работает на лидирующих на рынке версиях.

Компоненты Zabbix доступны и протестированы на следующих платформах:

Платформа	Сервер	Агент	Агент 2	Комментарий
Linux	x	x	x	
Windows	-	x	x	Zabbix агент поддерживается на всех версиях Windows для рабочих станций и серверов, начиная с XP (64-битная)/Server 2003.
Mac OS X	x	x	-	
IBM AIX	x	x	-	Zabbix агент не будет работать на платформах AIX ниже версий 6.1 TL07 / 7.1 TL01.
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Zabbix сервер/агент может работать также и на других Unix-подобных операционных системах.

Attention:
Zabbix отключает дампы памяти, если скомпилирован с шифрованием, и не запустится, если система не позволяет отключение дампов памяти.

Требуемое программное обеспечение

Zabbix использует современные веб-сервера, ведущие СУБД и язык сценариев PHP.

Стороннее внешнее программное обеспечение

Если программное обеспечение или библиотека указаны как обязательные, они строго необходимы. Необязательные компоненты требуются для поддержки некоторых определённых функций.

Программное обеспечение	Обязательный статус	Поддерживаемые версии	Комментарии
MySQL/Percona	One of	8.4.0-9.7.X	Требуется, если MySQL (или Percona) используется в качестве внутренней базы данных Zabbix. Требуется механизм InnoDB. Для сборки сервера/прокси рекомендуется использовать библиотеку C API (libmysqlclient). Настоятельно рекомендуется использовать в производственных средах только выпуски LTS. Выпуски с пометками «rolling release», «early access» и аналогичными не рекомендуются.

Программное обеспечение	Обязательный статус	Поддерживаемые версии	Комментарии
MariaDB		10.11.00-12.3.X	Требуется механизм InnoDB. Для сборки сервера/прокси рекомендуется использовать библиотеку MariaDB Connector/C. См. также: Возможные взаимоблокировки с MariaDB и Доступ к элементам веб-интерфейса с MariaDB 10.5.1-10.5.9. Настоятельно рекомендуется использовать в производственных средах только выпуски LTS. Выпуски с пометками «rolling release», «early access» и аналогичными не рекомендуются.
PostgreSQL		15.0-18.X	Требуется, если PostgreSQL используется в качестве внутренней базы данных Zabbix. В зависимости от размера установки может потребоваться увеличить значение свойства конфигурации PostgreSQL *work_mem* (значение по умолчанию — 4 МБ), чтобы объём памяти, используемый базой данных для определённой операции, был достаточным, а выполнение запроса не занимало слишком много времени.
TimescaleDB for PostgreSQL		2.20.X-2.29.X	Требуется, если TimescaleDB используется как расширение базы данных PostgreSQL. Убедитесь, что установлена TimescaleDB Community Edition, поддерживающая сжатие. Подробные сведения о совместимости версий PostgreSQL и TimescaleDB см. в документации TimescaleDB.
Elasticsearch		7.X	Требуется, если Elasticsearch используется в качестве внутренней базы данных Zabbix. Elasticsearch можно использовать только для хранения данных истории; он поддерживается только серверами Zabbix. Поддержка Elasticsearch в настоящее время является экспериментальной. См. также необходимое программное обеспечение для сервера/прокси.
ClickHouse		26.3.0.0-26.9.X.X	Требуется, если ClickHouse используется в качестве внутренней базы данных Zabbix для хранения данных истории (поддерживается только серверами Zabbix) или как источник данных телеметрии. Для использования ClickHouse в качестве источника данных телеметрии требуется ClickHouse версии 26.8.17.4 или более поздней. Системное время на сервере Zabbix и сервере ClickHouse также должно быть синхронизировано; в противном случае запросы телеметрии могут пропускать данные или возвращать одни и те же данные дважды.
SQLite	Optional	3.3.5-3.53.X	SQLite поддерживается только прокси Zabbix. Требуется, если SQLite используется в качестве базы данных прокси Zabbix.
smartmontools		7.1 or later	Требуется для агента Zabbix 2.
who			Требуется для плагина подсчёта пользователей.
dpkg			Требуется для плагина system.sw.packages.
pkgtool			Требуется для плагина system.sw.packages.
rpm			Требуется для плагина system.sw.packages.
pacman			Требуется для плагина system.sw.packages.
q applets			`qlist` и `qsize`, входящие в состав q applets, требуются для плагина system.sw.packages в Gentoo Linux.

Note:
Хотя Zabbix может работать с базами данных, доступными в операционных системах, для оптимальной работы рекомендуется использовать базы данных, установленные из официальных репозиториев разработчиков баз данных.

Веб-интерфейс

Минимальная поддерживаемая ширина экрана для веб-интерфейса Zabbix составляет 1200px.

Если программное обеспечение или библиотека указаны как обязательные, они строго необходимы. Необязательные компоненты требуются для поддержки некоторых определённых функций.

Программное обеспечение	Обязательность	Поддерживаемые версии	Комментарии
PHP	Да	8.2.0 - 8.5.X	
Apache	Один из	2.4 или более поздняя	
Nginx		1.20 или более поздняя	
MySQL	Один из	См. стороннее внешнее сопутствующее программное обеспечение	
PostgreSQL			
Расширения PHP			
mysqli	Да		Требуется, если MySQL используется в качестве внутренней базы данных сервера Zabbix.
pgsql			Требуется, если PostgreSQL используется в качестве внутренней базы данных сервера Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); требуется для поддержки пользовательских скриптов.
gd		2.0.28 или более поздняя	php-gd (если дистрибьютор предоставляет его в виде отдельного пакета); расширение PHP GD должно поддерживать изображения PNG (*--with-png-dir*), изображения JPEG (*--with-jpeg-dir*) и FreeType 2 (*--with-freetype-dir*). Для некоторых языков веб-интерфейса может потребоваться версия 2.3.0 или более поздняя, чтобы избежать возможного наложения текста на графиках.
libxml		2.6.15 или более поздняя	php-xml (если дистрибьютор предоставляет его в виде отдельного пакета)
xmlwriter			php-xmlwriter (если дистрибьютор предоставляет его в виде отдельного пакета)
xmlreader			php-xmlreader (если дистрибьютор предоставляет его в виде отдельного пакета)
ctype			php-ctype (*--enable-ctype*)
session			php-session (если дистрибьютор предоставляет его в виде отдельного пакета)
ldap	Нет		php-ldap; требуется для аутентификации LDAP.
openssl			php-openssl; требуется для аутентификации SAML.
gettext			php-gettext (*--with-gettext*); требуется для переводов.
cURL		7.19.4 или более поздняя	php-curl; требуется для MFA Duo Universal Prompt, аутентификации SMTP и тестирования источника данных телеметрии.

Сторонние библиотеки веб-интерфейса, поставляемые вместе с Zabbix:

Библиотека	Обязательность	Поставляемая версия	Комментарии
Библиотека JavaScript jQuery	Да	3.6.0	Библиотека JavaScript, упрощающая разработку для разных браузеров.
jQuery UI		1.12.1	Набор взаимодействий с пользовательским интерфейсом, эффектов, виджетов и тем, созданных на основе jQuery.
SAML PHP Toolkit		4.3.1	Инструментарий PHP, добавляющий поддержку аутентификации SAML 2.0 для входа в Zabbix.
Duo Universal PHP library		1.1.2	Библиотека PHP, добавляющая поддержку многофакторной аутентификации Duo Universal Prompt для Zabbix.
Symfony Yaml Component		5.1.0	Добавляет поддержку экспорта и импорта элементов конфигурации Zabbix в формате YAML.

Note:
Zabbix также может работать с более ранними версиями Apache, MySQL и PostgreSQL.

Attention:
Для шрифтов, отличных от используемого по умолчанию DejaVu, может потребоваться функция PHP imagerotate. Если она отсутствует, эти шрифты могут отображаться некорректно при отображении графика. Эта функция доступна только в том случае, если PHP скомпилирован со встроенной GD, что не относится к Debian и другим дистрибутивам.

Сторонние библиотеки, используемые для написания и отладки кода веб-интерфейса Zabbix:

Библиотека	Обязательность	Минимальная версия	Описание
Composer	Нет	2.4.1	Менеджер пакетов уровня приложения для PHP, предоставляющий стандартный формат управления зависимостями программного обеспечения PHP и необходимыми библиотеками.
PHPUnit		8.5.29	Среда модульного тестирования PHP для тестирования веб-интерфейса Zabbix.
SASS		3.4.22	Скриптовый язык препроцессора, который интерпретируется и компилируется в каскадные таблицы стилей (CSS).

Веб-браузер на стороне клиента

Cookies и JavaScript должны быть включены.

Поддерживаются последние стабильные версии Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari и Opera.

Warning:
Реализована политика одного источника для IFrames, что означает, что Zabbix веб-интерфейс нельзя поместить во фреймы на другом домене.

 Тем не менее, помещённые в Zabbix фреймы страницы будут иметь доступ к веб-интерфейсу Zabbix (посредством JavaScript), если помещённая во фрейм страница и веб-интерфейс Zabbix располагаются на одном домене. Такая станица как `http://secure-zabbix.com/cms/page.html`, при помещении в панели на `http://secure-zabbix.com/zabbix/`, будет иметь полный доступ к Zabbix посредством JS.

Сервер/прокси

Если программное обеспечение или библиотека указаны как обязательные, они строго необходимы. Необязательные компоненты нужны для поддержки некоторых функций.

Требование	Обязательный статус	Описание
libpcre2	Да	Библиотека PCRE2 необходима для поддержки совместимых с Perl регулярных выражений (PCRE). Поддерживается PCRE2 версии 10.x.
libevent		Необходима для межпроцессного взаимодействия. Версия 2.0.10 или выше.

Требование	Обязательный статус	Описание
libevent-pthreads		Необходима для межпроцессного взаимодействия.
libpthread		Необходима для поддержки мьютексов и блокировок чтения-записи (может входить в состав libc).
libresolv		Необходима для разрешения DNS-имен (может входить в состав libc).
libiconv		Необходима для преобразования кодировок и форматов текста (может входить в состав libc). Обязательна для сервера Zabbix в Linux.
libz		Необходима для поддержки сжатия.
libm		Математическая библиотека. Необходима только серверу Zabbix.
libmysqlclient	Одна из	Необходима, если используется MySQL.
libmariadb		Необходима, если используется MariaDB.
libpq5		Необходима, если используется PostgreSQL; версия *libpq5* должна совпадать с версией используемой базы данных PostgreSQL или быть выше нее.
libsqlite3		Необходима, если используется SQLite. Необходима только прокси Zabbix.
libOpenIPMI	Нет	Необходима для поддержки IPMI. Необходима только серверу Zabbix.
libssh2 или *libssh*		Необходима для проверок SSH. Версия 1.8.0 или выше (libssh2); 0.9.0 или выше (libssh).
libcurl		Необходима для следующих функций: - элементов данных веб-мониторинга, мониторинга VMware и HTTP-агента (для всех: версия 7.19.1 или выше); - элементов данных Zabbix агента web.page.* (версия 7.19.1 или выше; см. также требования к агенту); - аутентификации SMTP (Basic: версия 7.20.0 или выше; OAuth: версия 7.33 или выше; см. также требования к веб-интерфейсу); - Elasticsearch (версия 7.28.0 или выше); - элементов данных запроса телеметрии (версия 7.28.0 или выше). Для всех функций рекомендуется версия 7.28.0 или выше. Чтобы использовать обновленные возможности cURL для элементов данных `web.page.*`, перезапустите сервер/прокси Zabbix. Для аутентификации SMTP во время выполнения используйте пакет `libcurl-full`.
libxml2		Необходима для мониторинга VMware и предварительной обработки XML XPath.
net-snmp		Необходима для поддержки SNMP. Версия 5.3.0 или выше. Поддержка протоколов надежного шифрования (AES192/AES192C, AES256/AES256C) доступна начиная с библиотеки net-snmp 5.8; в системах на базе RHEL 8+ рекомендуется использовать net-snmp версии 5.8.15 или выше.
libunixodbc		Необходима для мониторинга баз данных.
libgnutls или *libopenssl*		Необходима при использовании шифрования. Минимальные версии: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Необходима для поддержки LDAP.
fping		Необходима для элементов данных проверки ICMP ping.
c-ares		Необходима для асинхронного разрешения DNS-имен, если Zabbix настроен с параметром `--with-ares`. В противном случае будет использоваться *libevent*. Минимальная версия: 1.16.0. Для функции кэширования DNS-запросов и переключения при сбое (кэш запросов включен по умолчанию) требуется c-ares версии 1.26.0 или выше. В системах, где c-ares предоставляется пакетом ОС, убедитесь, что установленная библиотека времени выполнения предоставляет функции разрешения имен, необходимые для кэширования DNS-запросов и переключения при сбое.
g++		Компилятор C++. Необходим для компиляции прокси Zabbix с параметром `--with-apm`.

Требование	Обязательный статус	Описание
gRPC		Необходим для поддержки APM на прокси Zabbix. Необходимые библиотеки: *grpc++*, *grpc*, *gpr*. Для компиляции прокси Zabbix также необходима утилита `grpc_cpp_plugin`. Требуемая версия доступна в стандартных репозиториях пакетов следующих операционных систем: - Ubuntu 23.04 или выше; - Debian 12 (Bookworm) или выше; - Alpine Linux 3.15 или выше; - RHEL, Rocky Linux, CentOS Stream 9 или выше (только из репозитория EPEL); - Fedora X или выше. В более старых версиях операционных систем *gRPC* необходимо собирать из исходного кода.
protobuf		Необходим для поддержки APM на прокси Zabbix. Для компиляции прокси Zabbix также необходим компилятор `protoc`.
Abseil		Необходим для поддержки APM на прокси Zabbix. Необходимые библиотеки: *absl_synchronization*, *absl_base*. Обычно устанавливается вместе с *gRPC*.

Агент

Requirement	Mandatory status	Description
libpcre2	Yes	Требуется библиотека PCRE2 для поддержки Perl Compatible Regular Expression (PCRE). Поддерживается PCRE2 v10.x. Требуется для мониторинга журналов. Также требуется в Windows.
libpthread		Требуется для поддержки mutex и read-write lock (может быть частью libc). В Windows не требуется.
libresolv		Требуется для разрешения DNS (может быть частью libc). В Windows не требуется.
libiconv		Требуется для преобразования текстовой кодировки/формата в UTF-8 в элементах данных журнала, содержимого файлов, регулярных выражений файлов и элементах данных regmatch (может быть частью libc). В Windows не требуется.
libgnutls or *libopenssl*	No	Требуется при использовании шифрования. Минимальные версии: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 В Microsoft Windows требуется OpenSSL 1.1.1 или более поздняя версия.
libldap		Требуется, если используется LDAP. Не поддерживается в Windows.
libcurl		Требуется для расширенной поддержки элементов данных Zabbix agent web.page.*. Без libcurl доступна базовая функциональность (например, web.page.get[http://example.com]). С libcurl агент поддерживает дополнительные возможности, такие как HTTP URL с учетными данными (например, http://user:password@example.com) и HTTPS URL. Требуется версия 7.19.1 или выше (рекомендуется 7.28.0 или выше). Чтобы использовать обновленные возможности cURL, перезапустите агент Zabbix.
libmodbus		Требуется только при использовании мониторинга Modbus. Версия 3.0 или выше.

Агент 2

Requirement	Mandatory status	Description
Go	Yes	Требуется для сборки Zabbix agent 2 и его плагинов из исходного кода. Поддерживается Go 1.25 или более поздняя версия. См. go.dev для инструкций по установке. Библиотеки Go, используемые Zabbix agent 2 и его плагинами, перечислены в репозитории Zabbix Git (библиотеки, помеченные как `indirect` в репозитории, являются зависимостями других требуемых библиотек): - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	Для поддержки Perl Compatible Regular Expression (PCRE) требуется библиотека PCRE2. Поддерживается PCRE2 v10.x. Требуется для мониторинга логов. Также требуется в Windows.
libopenssl	No	Требуется при использовании шифрования. На платформах UNIX требуется OpenSSL 1.0.1 или более поздняя версия. Библиотека OpenSSL должна иметь включенную поддержку PSK. LibreSSL не поддерживается. В системах Microsoft Windows требуется OpenSSL 1.1.1 или более поздняя версия.

Веб-сервис

Для создания отчётов по расписанию с использованием веб-сервиса Zabbix поддерживается последняя стабильная версия Google Chrome.

Версия Go, требуемая для построения веб-сервиса, совпадает с той, которая используется для Zabbix агента 2.

Java gateway

Если вы получили Zabbix из репозитория исходного кода или архива, необходимые зависимости уже включены в дерево исходного кода.

Если вы получили Zabbix из пакета вашего дистрибутива, необходимые зависимости уже предоставляются системой управления пакетами.

В обоих описанных выше случаях программное обеспечение готово к использованию, и дополнительные загрузки не требуются.

Однако если вы хотите предоставить собственные версии этих зависимостей (например, если вы готовите пакет для какого-либо дистрибутива Linux), ниже приведён список версий библиотек, с которыми, как известно, работает Java gateway. Zabbix также может работать с другими версиями этих библиотек.

В следующей таблице перечислены JAR-файлы, которые в настоящее время входят в состав Java gateway в исходном коде:

Библиотека	Обязательность	Включённая версия	Комментарии
android-json	Да	4.3r1	JSON (JavaScript Object Notation) — это облегчённый формат обмена данными. Это совместимая с `org.json` реализация для Android, извлечённая из Android SDK.
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java gateway можно собрать с использованием Oracle Java или свободно распространяемого OpenJDK (версии 8 или новее). Пакеты, предоставляемые Zabbix, компилируются с использованием OpenJDK. В следующей таблице перечислены пакеты OpenJDK, используемые дистрибутивами для сборки пакетов Zabbix:

Дистрибутив	Пакет OpenJDK
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Адаптер моста

Адаптер моста Zabbix не имеет особых требований.

Порты по умолчанию

В следующем списке указаны порты, которые компоненты Zabbix могут использовать в конфигурации по умолчанию:

Компонент Zabbix	Протокол и порт	Соединение инициируется
Агент Zabbix	TCP 10050	Сервер Zabbix или прокси Zabbix (только пассивные проверки)
Агент Zabbix 2	TCP 10050	Сервер Zabbix или прокси Zabbix (только пассивные проверки)
Сервер Zabbix	TCP 10051	Активные агенты Zabbix, активный прокси Zabbix, Zabbix sender, веб-интерфейс Zabbix
Прокси Zabbix	TCP 10051	Активные агенты Zabbix, Zabbix sender, сервер Zabbix (только пассивный прокси)
Java gateway Zabbix	TCP 10052	Сервер Zabbix или прокси

Компонент Zabbix	Протокол и порт	Соединение инициируется
Веб-сервис Zabbix	TCP 10053	Сервер Zabbix
MCP-сервер Zabbix	TCP 8443	Клиенты MCP (агенты ИИ)
Веб-интерфейс Zabbix	TCP 80	Веб-браузеры, веб-сервис Zabbix (плановые отчеты), клиенты API
	TCP 443	
SNMP-ловушки (snmptrapd)	UDP 162	Устройства SNMP
Адаптер моста Zabbix	TCP 10054	Сервер Zabbix

Warning:
Открывайте только порты, необходимые для фактически используемых компонентов, и ограничивайте каждое правило брандмауэра конкретными узлами сети, которым требуется подключение (например, разрешайте порт 10050 только для сервера Zabbix или прокси, а не для любого адреса). Не открывайте порты Zabbix напрямую в интернет.
 Перед открытием портов защитите установку: включите шифрование между компонентами, ограничьте разрешенные узлы сети в конфигурационных файлах компонентов и ознакомьтесь с рекомендациями по обеспечению безопасности.

Размер базы данных

Конфигурационные данные Zabbix требуют фиксированного объема дискового пространства и не сильно увеличиваются.

Размер базы данных Zabbix в основном зависит от следующих переменных, которые определяют объем хранимых исторических данных:

- Количество обрабатываемых значений в секунду

Это среднее количество новых значений, которое сервер Zabbix получает каждую секунду. Например, если у нас есть 3000 элементов данных для мониторинга с интервалом обновления 60 секунд, количество значений в секунду рассчитывается как 3000/60 = **50**.

Это означает, что каждую секунду в базу данных Zabbix добавляется 50 новых значений.

- Настройки housekeeper для истории

Zabbix хранит значения в течение фиксированного периода времени, обычно нескольких недель или месяцев. Каждое новое значение требует определенного объема дискового пространства для данных и индекса.

Итак, если мы хотим хранить историю за 30 дней и получаем 50 значений в секунду, общее количество значений составит примерно (**30***24*3600)* **50** = 129.600.000, или около 130M значений.

В зависимости от используемого движка базы данных, типа получаемых значений (числа с плавающей точкой, целые числа, строки, журналы и т. д.) объем дискового пространства, необходимый для хранения одного значения, может варьироваться от 40 байт до сотен байт. Обычно для числовых элементов данных это около 90 байт на значение[2]. В нашем случае это означает, что для 130M значений потребуется 130M * 90 байт = **10.9GB** дискового пространства.

Note:
Размер значений текстовых/журнальных элементов данных невозможно точно предсказать, но можно ожидать около 500 байт на значение.

- Настройки housekeeper для трендов

Zabbix хранит набор значений max/min/avg/count за 1 час для каждого элемента данных в таблице **trends**. Эти данные используются для построения трендов и графиков за длительные периоды. Период в один час нельзя настраивать.

База данных Zabbix, в зависимости от типа базы данных, требует около 90 байт на каждый итог. Предположим, что мы хотим хранить данные трендов за 5 лет. Для значений 3000 элементов данных потребуется 3000*24*365* **90** = **2.2GB** в год, или **11GB** за 5 лет.

- Настройки housekeeper для событий

Каждое событие Zabbix требует примерно 250 байт дискового пространства[1]. Трудно оценить количество событий, ежедневно генерируемых Zabbix. В худшем случае можно предположить, что Zabbix генерирует одно событие в секунду.

Для каждого восстановленного события создается запись event_recovery. Обычно большинство событий будет восстановлено, поэтому можно предположить одну запись event_recovery на событие. Это означает дополнительные 80 байт на событие.

При необходимости события могут иметь теги, при этом каждая запись тега требует примерно 100 байт дискового пространства[1]. Количество тегов на событие (#tags) зависит от конфигурации. Следовательно, для каждого события потребуется дополнительно #tags * 100 байт дискового пространства.

Это означает, что если мы хотим хранить события за 3 года, потребуется 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B дискового пространства[2].

Note:
[1] Больше при наличии не-ASCII имен событий, тегов и значений.
 [2] Оценки размера основаны на MySQL и могут отличаться для других баз данных.

В таблице приведены формулы, которые можно использовать для расчета объема дискового пространства, необходимого для системы Zabbix:

Параметр	Формула для требуемого объема дискового пространства (в байтах)
Конфигурация Zabbix	Фиксированный размер. Обычно 10MB или меньше.
История	days*(items/refresh rate)*24*3600*bytes items : количество элементов данных days : количество дней хранения истории refresh rate : средний интервал обновления элементов данных bytes : количество байт, необходимое для хранения одного значения, зависит от движка базы данных, обычно ~90 байт.
Тренды	days*(items/3600)*24*3600*bytes items : количество элементов данных days : количество дней хранения истории bytes : количество байт, необходимое для хранения одного тренда, зависит от движка базы данных, обычно ~90 байт.
События	days*events*24*3600*bytes events : количество событий в секунду. В худшем случае — одно (1) событие в секунду. days : количество дней хранения истории bytes : количество байт, необходимое для хранения одного события, зависит от движка базы данных, обычно ~330 + среднее количество тегов на событие * 100 байт.

Итак, общий требуемый объем дискового пространства можно рассчитать как:

Конфигурация + История + Тренды + События

Дисковое пространство НЕ будет использоваться сразу после установки Zabbix. Размер базы данных будет расти, а затем в какой-то момент перестанет увеличиваться, что зависит от настроек housekeeper.

Синхронизация времени

Очень важно иметь точное системное время на сервере с запущенным Zabbix. ntpd [en] — один из самых популярных демонов синхронизации времени узла с временем на остальных серверах. Настоятельно рекомендуется поддерживать синхронизированное время на всех системах, где работают компоненты Zabbix.

Новое и примечания по обновлению: 8.0

Zabbix 8.0.0 является продолжением Zabbix 7.4.0 и включает новые функции и улучшения.

Также на этой странице:

- Новое в шаблонах, включая новые шаблоны и изменения в существующих шаблонах.
- Примечания по обновлению, включая критические изменения.

См. также: Изменения API.

Примечания к обновлению

Эти примечания предназначены для обновления с Zabbix 7.4.x до Zabbix 8.0.0.

Все примечания сгруппированы в:

- **Критические изменения** - изменения, которые могут нарушить работу существующих установок, и другая важная информация, связанная с процессом обновления
- **Другое** - вся остальная информация, описывающая изменения в функциональности Zabbix

См. также:

- Процедура обновления — вся необходимая информация об обновлении с версий до Zabbix 8.0.0;
- Обновление кластера HA — инструкции по обновлению серверов в кластере **высокой доступности** (HA).

Изменения, несовместимые с предыдущими версиями

Другое

Новые возможности и примечания по обновлению: 8.0.x

На этой странице представлены все новые возможности, включенные в минорные выпуски основной версии Zabbix. Также приведены изменения шаблонов и примечания по обновлению.

См. также раздел Новые возможности основной версии.

Что нового в Zabbix 8.0.1 Эта версия еще не выпущена.

См. примечания к обновлению для этой версии.

Примечания к обновлению

Zabbix appliance

Обзор Устройство Zabbix позволяет мгновенно развернуть сервер Zabbix и веб-интерфейс вместо их ручной настройки или повторного использования существующего сервера для Zabbix.

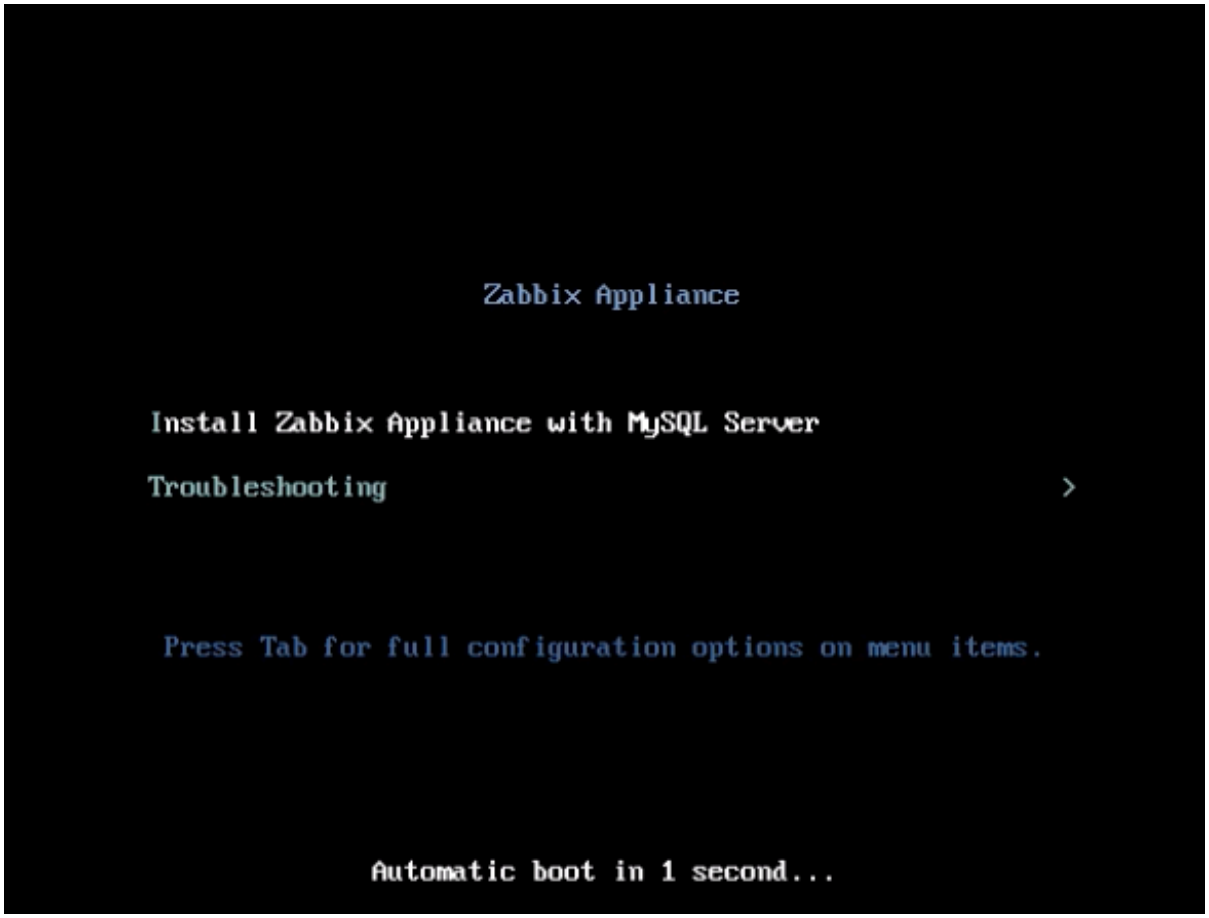
Устройство для Zabbix 8.0 основано на производных Red Hat Enterprise Linux 10, таких как AlmaLinux или Rocky Linux, и содержит предварительно настроенный сервер Zabbix, работающий на MySQL, а также веб-интерфейс, работающий на веб-сервере NGINX.

> **Attention:**
> Это устройство предназначено для оценки возможностей Zabbix. Не рекомендуется использовать его в серьезных производственных средах.

Образы устройства доступны для скачивания в следующих форматах:

- Установочный CD/DVD (.iso)
- VMware (.vmx) - см. примечания
- Формат открытой виртуализации (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - см. примечания
- KVM, Parallels, QEMU, USB-накопитель, VirtualBox, Xen (.raw) - см. примечания
- KVM, QEMU (.qcow2)

Меню загрузки установочного CD/DVD Zabbix:



Быстрый старт Предварительные требования

1. Убедитесь, что хост-машина имеет достаточно ресурсов для соответствия системным требованиям виртуальной машины:

- *Оперативная память*: 4 ГБ
- *Дисковое пространство*: для виртуальной машины должно быть выделено не менее 8 ГБ
- *CPU*: минимум 2 ядра

2. Если программное обеспечение для виртуализации, необходимое для загрузки образа appliance, еще не установлено, установите его (например, VirtualBox).
3. Загрузите appliance в формате, поддерживаемом вашим программным обеспечением для виртуализации.
4. Проверьте сетевые настройки, чтобы убедиться, что DHCP включен на хост-машине.

Установка

1. Загрузите виртуальную машину appliance с образа, который был скачан. Инструкции см. в документации к вашему программному обеспечению для виртуализации, например, в документации VirtualBox.
2. Настройте сетевые параметры виртуальной машины так, чтобы обеспечить доступ из браузера на узле сети. Это можно сделать, включив *режим Bridged*.
3. Войдите в виртуальную машину, используя системные учетные данные по умолчанию.
4. Чтобы получить IP-адрес, выполните на виртуальной машине следующую команду:

```
ip addr show
```

5. Откройте браузер на узле сети и укажите IP-адрес, который appliance получил по DHCP.
6. Войдите в Zabbix, используя учетные данные по умолчанию, и начните мониторинг.

Конфигурация В этом разделе описаны часто требуемые настройки конфигурации по умолчанию, а также доступные варианты их настройки.

Учетные данные

Система

- Имя пользователя: `root`
- Пароль: `zabbix`

Веб-интерфейс Zabbix

- Имя пользователя: `Admin`
- Пароль: `zabbix`

После входа в систему можно изменить пароль по умолчанию в настройках профиля пользователя или создать новых пользователей и удалить пользователя по умолчанию.

База данных

Пароли для всех пользователей базы данных генерируются случайным образом во время установки. В базе данных определены следующие пользователи:

Root:

- Имя пользователя: `root`
- Пароль: пароль хранится в файле `/root/.my.cnf` Вводить пароль под учетной записью root не требуется.

Сервер Zabbix:

- Имя пользователя: `zabbix_srv`
- Пароль: пароль хранится в файле конфигурации сервера

Веб-интерфейс Zabbix:

- Имя пользователя: `zabbix_web`
- Пароль: пароль хранится в файле конфигурации веб-интерфейса

Чтобы изменить пароль пользователя базы данных, измените его в MySQL и в соответствующем файле конфигурации.

Доступ к веб-интерфейсу

Доступ к веб-интерфейсу Zabbix можно получить по адресу *http://<IP-адрес виртуальной машины>*.

По умолчанию доступ разрешен из любого места. Чтобы ограничить доступ, измените файл `/etc/nginx/conf.d/zabbix.conf`. После сохранения отредактированного файла перезапустите NGINX, войдя по SSH от имени пользователя root и выполнив:

```
systemctl restart nginx
```

Статический IP-адрес

По умолчанию устройство использует DHCP для получения IP-адреса. Чтобы задать статический IP-адрес:

1. Войдите как пользователь root.
2. Выполните следующие команды, заменив значения на ваш собственный IP-адрес:

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # IP-   /CIDR-
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # IP-
nmcli con mod eth0 ipv4.dns 8.8.8.8    # IP-    DNS-
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

В качестве альтернативы можно открыть файл `/etc/NetworkManager/system-connections/eth0.nmconnection` и добавить следующие строки:

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

После сохранения измененного файла выполните команды:

```
nmcli con reload
nmcli con up eth0
```

Чтобы проверить, что изменения конфигурации сети были применены, выполните следующую команду:

```
nmcli dev show eth0
```

Конфигурация брандмауэра

Для управления настройками брандмауэра устройство использует iptables с предопределенными правилами:

- Открыть порт SSH (22 TCP)
- Открыть порты Zabbix agent (10050 TCP) и Zabbix trapper (10051 TCP)
- Открыть порты HTTP (80 TCP) и HTTPS (443 TCP)
- Открыть порт SNMP trap (162 UDP)
- Разрешить исходящие соединения к порту NTP (123 UDP)
- Ограничить пакеты ICMP до 5 пакетов в секунду
- Отклонять все остальные входящие соединения

Чтобы открыть дополнительные порты, измените файл `/etc/sysconfig/iptables` и перезагрузите правила брандмауэра:

```
systemctl reload iptables
```

Репозитории

В Zabbix appliance используется пакет *zabbix-release* из репозитория Zabbix. Репозитории настраиваются в каталоге `/etc/yum.repos.d/*`.

Часовой пояс

По умолчанию appliance использует UTC для системных часов. Чтобы изменить часовой пояс, скопируйте соответствующий файл из `/usr/share/zoneinfo` в `/etc/localtime`, например:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

Note:
Часовой пояс веб-интерфейса Zabbix задается отдельно и может быть изменен в настройках веб-интерфейса. Часовой пояс по умолчанию для веб-интерфейса Zabbix — Europe/Riga.

Расположение файлов

- Файлы конфигурации сервера и агента Zabbix находятся в `/etc/zabbix`.
- Файлы журналов сервера и агента Zabbix находятся в `/var/log/zabbix`.
- Файл конфигурации веб-интерфейса Zabbix находится в `/etc/zabbix/web`.
- Веб-интерфейс Zabbix находится в `/usr/share/zabbix`.
- Домашний каталог пользователя *zabbix* находится в `/var/lib/zabbix`.

Системные службы

Доступны службы systemd. Чтобы увидеть список служб Zabbix, выполните следующую команду на виртуальной машине:

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

Примечания, специфичные для формата изображения VMware

Образы в формате *vmdk* можно использовать непосредственно в продуктах VMware Player, Server и Workstation. Для использования в ESX, ESXi и vSphere их необходимо преобразовать с помощью VMware vCenter Converter (для загрузки требуется аутентификация). При использовании VMware vCenter Converter могут возникнуть проблемы с гибридным сетевым адаптером. В этом случае можно попробовать указать адаптер E1000 в процессе преобразования. Кроме того, после завершения преобразования можно удалить существующий адаптер и добавить адаптер E1000.

HDD/flash image (raw)

Чтобы загрузить образ, выполните:

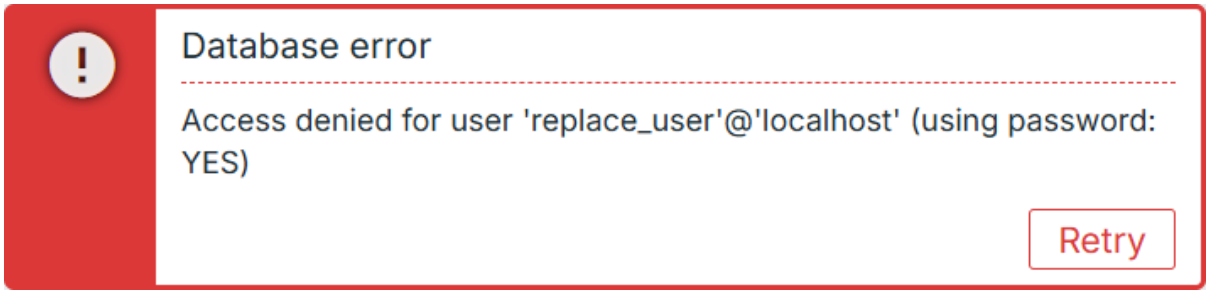
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Замените `/dev/sdc` на путь к устройству вашего диска Flash/HDD.

Hyper-V

Если appliance не удаётся запустить в Hyper-V, попробуйте нажать **Ctrl+Alt+F2**, чтобы переключиться на сеанс TTY.

Устранение неполадок Если при попытке входа во веб-интерфейс вы сталкиваетесь с сообщением об ошибке *Access denied for user 'replace_user'@'localhost' (using password: YES)*, это может означать, что установка все еще выполняется.



Если ошибка сохраняется после ожидания в течение нескольких минут или вы замечаете любое другое неожиданное поведение, это, вероятно, означает, что процесс установки не был успешно завершен. В этом случае мы рекомендуем удалить текущий appliance и развернуть его заново, следуя тем же инструкциям по установке. Обычно этот шаг устраняет проблему.

Обратите внимание, что попытки вручную исправить поврежденную установку не рекомендуются, так как это может привести к дополнительным осложнениям.

Процессы

Обзор Zabbix состоит из нескольких процессов, которые работают вместе для мониторинга систем и отправки уведомлений. Каждый процесс выполняет свою определенную роль в инфраструктуре мониторинга.

Основные процессы

- Сервер - центральный процесс, который выполняет опрос и прием данных, вычисляет триггеры и отправляет уведомления пользователям.
- Агент - собирает данные с отслеживаемых узлов сети.
- Агент 2 - агент нового поколения с архитектурой плагинов и расширенными возможностями.
- Прокси - собирает данные мониторинга с отслеживаемых устройств от имени сервера Zabbix и отправляет информацию на централизованный сервер Zabbix в распределенной конфигурации.
- Java gateway - обеспечивает мониторинг приложений Java.
- Сервер MCP - позволяет агентам ИИ работать с Zabbix с использованием протокола Model Context Protocol.

Для удобного доступа к Zabbix из любого места и с любой платформы предоставляется веб-интерфейс. Веб-интерфейс является частью сервера Zabbix и обычно (но не обязательно) работает на том же физическом компьютере, что и сервер.

Вся информация о конфигурации, а также данные, собранные Zabbix, хранятся в базе данных.

Заголовок Заголовок присутствует во всех сообщениях запроса и ответа между компонентами Zabbix.
Он необходим для определения длины сообщения, того, сжато оно или нет, а также является ли оно большим пакетом.

Протокол обмена данными Zabbix имеет ограничение на размер пакета 1 ГБ на одно соединение. Ограничение в 1 ГБ применяется как к длине полученных данных пакета, так и к длине несжатых данных.

При отправке конфигурации на прокси Zabbix ограничение на размер пакета увеличивается до 4 ГБ, чтобы обеспечить синхронизацию больших конфигураций. Когда длина данных до сжатия превышает 4 ГБ, сервер Zabbix автоматически начинает использовать формат большого пакета (флаг 0x04), который увеличивает ограничение на размер пакета до 16 ГБ.

Обратите внимание, что хотя формат большого пакета можно использовать для отправки любых данных, в настоящее время только синхронизатор конфигурации прокси Zabbix может обрабатывать пакеты размером более 1 ГБ.

Структура заголовка Заголовок состоит из четырех полей. Все числа в заголовке представлены в формате little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" или 5A 42 58 44
<FLAGS>	1	1	Флаги протокола: 0x01 - протокол обмена данными Zabbix 0x02 - сжатие 0x04 - большой пакет

Field	Size	Size (large packet)	Description
<DATALEN>	4	8	Длина данных.
<RESERVED>	4	8	При использовании сжатия (флаг 0x02) - длина несжатых данных При отсутствии сжатия - 00 00 00 00

Примеры заголовка Ниже приведены несколько фрагментов кода, показывающих, как добавить заголовок протокола Zabbix к данным, которые вы хотите отправить, чтобы получить пакет, который следует отправить в Zabbix, чтобы он был корректно интерпретирован. В этих фрагментах кода предполагается, что размер данных не превышает 1 ГБ, поэтому формат большого пакета не используется.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

или

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

или

```
sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }

    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;
```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

или

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

Сервер

Обзор Сервер Zabbix — это центральный процесс программного обеспечения Zabbix.

Сервер выполняет опрос и прием данных, вычисляет триггеры, отправляет уведомления пользователям. Это центральный компонент, которому агенты Zabbix и прокси передают данные о доступности и целостности систем. Сервер также может удаленно проверять сетевые службы (например, веб-серверы и почтовые серверы) с помощью простых проверок служб.

Сервер является центральным хранилищем, в котором сохраняются все конфигурационные, статистические и эксплуатационные данные, и именно он в Zabbix активно оповещает администраторов при возникновении проблем в любых контролируемых системах.

Работа базового сервера Zabbix разделена на три отдельных компонента: сервер Zabbix, веб-интерфейс и хранилище базы данных.

Вся информация о конфигурации Zabbix хранится в базе данных, с которой взаимодействуют и сервер, и веб-интерфейс. Например, когда вы создаете новый элемент данных с помощью веб-интерфейса (или API), он добавляется в таблицу items в базе данных. Затем примерно раз в минуту сервер Zabbix запрашивает таблицу items, чтобы получить список активных элементов данных, который затем сохраняется в кэше внутри сервера Zabbix. Именно поэтому для отображения любых изменений, внесенных в веб-интерфейсе Zabbix, в разделе последних данных может потребоваться до двух минут.

См. также:

- Настройка высокой доступности для серверов Zabbix
- Параметры конфигурации: сервер Zabbix

Запущенный сервер Если установлен из пакета

Сервер Zabbix работает как фоновый процесс. Сервер можно запустить, выполнив:

```
systemctl start zabbix-server
```

Это будет работать в большинстве систем GNU/Linux. В других системах может потребоваться выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния используйте следующие команды:

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

Запуск вручную

Если указанный выше способ не работает, необходимо запустить его вручную. Найдите путь к двоичному файлу zabbix_server и выполните:

```
zabbix_server
```

С Zabbix server можно использовать следующие параметры командной строки:

Parameter	Description
-c --config <config-file>	Путь к файлу конфигурации.
-f --foreground	Запустить Zabbix server в режиме переднего плана.
-R --runtime-control <option>	Выполнить административные функции.
-T --test-config	Проверить файл конфигурации и выйти.
-h --help	Вывести справочную информацию и выйти.
-V --version	Показать номер версии.

Примеры запуска Zabbix server с параметрами командной строки:

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

Управление во время выполнения

Параметры управления во время выполнения:

Параметр	Описание	Цель
config_cache_reload	Перезагрузить кэш конфигурации. Игнорируется, если кэш в данный момент загружается.	
history_cache_diff[=<target>]	Отличия кэш истории для элемента данных, указанного по его ID. Затрагивает все значения элемента данных, кроме первого и последнего.	**target** - ID элемента данных
diaginfo[=<section>]	Собрать диагностическую информацию в файле журнала сервера.	historycache - статистика кэша истории; valuecache - статистика кэша значений; preprocessing - статистика менеджера предварительной обработки; alerting - статистика менеджера оповещений; lld - статистика менеджера LLD; locks - список мьютексов (пусто в системах *BSD*); connector - статистика коннекторов с наибольшей очередью.
ha_status	Записать в журнал состояние кластера высокой доступности (HA).	
ha_remove_node=<target>	Удалить узел высокой доступности (HA), указанный по его имени или ID. Обратите внимание, что активные узлы и узлы в режиме ожидания удалить нельзя.	**target** - имя или ID узла (можно получить, выполнив ha_status).
ha_set_failover_delay=<delay>	Установить задержку переключения высокой доступности (HA). Поддерживаются суффиксы времени, например 10s, 1m.	

Параметр	Описание	Цель
`proxy_config_cache_reload[=<target>]`	Перезагрузить конфигурации прокси.	**target** - разделенный запятыми список имен прокси. Если цель не указана, конфигурация перезагружается для всех прокси.
`secrets_reload`	Перезагрузить секреты из Vault.	
`service_cache_reload`	Перезагрузить кэш менеджера служб.	
`snmp_cache_reload`	Перезагрузить кэш SNMP — очистить свойства движка SNMP (время работы движка, количество запусков движка, ID движка, учетные данные) для всех узлов сети. Используйте этот параметр для принудительной полной очистки кэша при устранении проблем с SNMP.	
`housekeeper_execute`	Запустить процедуру очистки. Игнорируется, если процедура очистки уже выполняется.	
`trigger_housekeeper_execute`	Запустить процедуру очистки триггеров. Игнорируется, если процедура очистки триггеров уже выполняется.	
`log_level_increase[=<target>]`	Повысить уровень журнала; если цель не указана, это затрагивает все процессы. Не поддерживается в системах *BSD*.	**process type** - все процессы указанного типа (например, `poller`). См. все типы процессов сервера. **process type,N** - тип и номер процесса (например, `poller,3`). **pid** - идентификатор процесса (1 до 65535). Для больших значений укажите цель в формате «тип процесса,N».
`log_level_decrease[=<target>]`	Понизить уровень журнала; если цель не указана, это затрагивает все процессы. Не поддерживается в системах *BSD*.	
`prof_enable[=<target>]`	Включить профилирование. Если цель не указана, это затрагивает все процессы. Включенное профилирование предоставляет сведения обо всех rwlock/мьютексах с указанием имени функции.	**process type** - все процессы указанного типа (например, `history syncer`) Поддерживаемые типы процессов в качестве целей профилирования: `alerter`, `alert manager`, `availability manager`, `configuration syncer`, `discovery manager`, `escalator`, `history poller`, `history syncer`, `housekeeper`, `http poller`, `icmp pinger`, `ipmi manager`, `ipmi poller`, `java poller`, `lld manager`, `lld worker`, `odbc poller`, `poller`, `preprocessing manager`, `proxy poller`, `self-monitoring`, `service manager`, `snmp trapper`, `task manager`, `timer`, `trapper`, `unreachable poller`, `vmware collector`. **process type,N** - тип и номер процесса (например, `history syncer,1`). **pid** - идентификатор процесса (1 до 65535). Для больших значений укажите цель в формате «тип процесса,N». **scope** - `rwlock`, `mutex`, `processing` можно использовать с типом и номером процесса (например, `history syncer,1,processing`) или со всеми процессами указанного типа (например, `history syncer,rwlock`).

Параметр	Описание	Цель
prof_disable[=<target>]	Отключить профилирование. Если цель не указана, это затрагивает все процессы.	**process type** - все процессы указанного типа (например, `history syncer`). Поддерживаемые типы процессов в качестве целей профилирования: см. `prof_enable`. **process type,N** - тип и номер процесса (например, `history syncer,1`). **pid** - идентификатор процесса (1 до 65535). Для больших значений укажите цель в формате «тип процесса,N».

Когда тип процесса указан как цель для управления во время выполнения, команда применяется только к выполняющимся процессам этого типа. Если в данный момент не выполняется ни один процесс указанного типа, команда возвращает ошибку:

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

Пример использования управления во время выполнения для перезагрузки кэша конфигурации сервера:

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

Примеры использования управления во время выполнения для перезагрузки конфигурации прокси:

```
###                        :
zabbix_server -R proxy_config_cache_reload

###                  Proxy1   Proxy2:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Пример использования управления во время выполнения для очистки кэша истории для элемента данных:

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

Примеры использования управления во время выполнения для сбора диагностической информации:

```
###                                          :
zabbix_server -R diaginfo

###                                  :
zabbix_server -R diaginfo=historycache
```

Пример использования управления во время выполнения для перезагрузки кэша SNMP:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Когда интерфейс SNMPv3 обновляется через веб-интерфейс Zabbix, Zabbix в большинстве случаев автоматически перезагружает новые учетные данные SNMPv3 для этого интерфейса; используйте `-R snmp_cache_reload` только если опрос по-прежнему не выполняется после изменения учетных данных (например, из-за несоответствий engineBoots/engineID или устройств, не соответствующих RFC), либо когда требуется принудительно очистить глобальный кэш SNMP для устранения неполадок.

Пример использования управления во время выполнения для запуска housekeeper:

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

Примеры использования управления во время выполнения для изменения уровня журналирования:

```
###                            :
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

###                              poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

###                           PID 1234:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

```
###                              http poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Пример установки задержки переключения при отказе HA на минимальное значение 10 секунд:

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

Пользователь процесса

Сервер Zabbix предназначен для запуска от пользователя, не являющегося root. Он будет работать от того пользователя, от которого был запущен, если это не root. Поэтому вы можете запускать сервер от любого пользователя, не являющегося root, без каких-либо проблем.

Если вы попытаетесь запустить его от 'root', он переключится на жестко заданного пользователя 'zabbix', который должен быть создан в вашей системе. Вы можете запускать сервер от 'root' только в том случае, если соответствующим образом измените параметр 'AllowRoot' в файле конфигурации сервера.

Если сервер Zabbix и агент запущены на одном и том же компьютере, рекомендуется использовать для запуска сервера другого пользователя, чем для запуска агента. В противном случае, если оба процесса запущены от одного и того же пользователя, агент может получить доступ к файлу конфигурации сервера, и любой пользователь уровня Admin в Zabbix может довольно легко получить, например, пароль базы данных.

Файл конфигурации

Путь по умолчанию к файлу конфигурации сервера зависит от операционной системы и способа установки Zabbix:

- При установке из исходных кодов: `/usr/local/etc/zabbix_server.conf` или путь, заданный переменными `--sysconfdir` или `--prefix` на этапе компиляции.
- При установке из пакетов и из appliance: `/etc/zabbix/zabbix_server.conf`.

Подробную информацию о настройке `zabbix_server` см. в разделе с параметрами файла конфигурации.

Скрипты запуска

Скрипты используются для автоматического запуска/остановки процессов Zabbix при включении/выключении системы. Скрипты располагаются в директории misc/init.d.

Типы процессов и потоков сервера

- `agent poller` - асинхронный процесс опроса для пассивных проверок с рабочим потоком;
- `alert manager` - менеджер очереди оповещений;
- `alert syncer` - процесс записи оповещений в базу данных;
- `alerter` - процесс отправки уведомлений;
- `availability manager` - процесс обновления доступности узлов сети;
- `browser poller` - процесс опроса для проверок элементов данных браузера;
- `configuration syncer` - процесс управления кэшем данных конфигурации в памяти;
- `configuration syncer worker` - процесс разрешения и синхронизации значений пользовательских макросов в именах элементов данных;
- `connector manager` - процесс управления коннекторами;
- `connector worker` - процесс обработки запросов от менеджера коннекторов;
- `discovery manager` - процесс управления обнаружением устройств;
- `discovery worker` - процесс обработки задач обнаружения от менеджера обнаружения;
- `escalator` - процесс эскалации действий;
- `ha manager` - процесс управления высокой доступностью;
- `history poller` - процесс обработки вычисляемых проверок, требующих подключения к базе данных;
- `history syncer` - процесс записи истории в базу данных;
- `housekeeper` - процесс удаления устаревших данных (истории и трендов элементов данных, пользовательских сеансов, событий и т. д.), а также данных, оставшихся после удаления объектов;
- `http agent poller` - асинхронный процесс опроса для HTTP-проверок с рабочим потоком;
- `http poller` - процесс опроса веб-мониторинга;
- `icmp pinger` - процесс опроса для проверок icmpping;
- `internal poller` - процесс опроса для внутренних проверок;
- `ipmi manager` - менеджер процесса опроса IPMI;
- `ipmi poller` - процесс опроса для проверок IPMI;
- `java poller` - процесс опроса для проверок Java;
- `lld manager` - процесс управления задачами низкоуровневого обнаружения;
- `lld worker` - рабочий процесс задач низкоуровневого обнаружения;
- `odbc poller` - процесс опроса для проверок ODBC;

- poller - обычный процесс опроса для пассивных проверок;
- preprocessing manager - менеджер задач предварительной обработки с рабочими потоками предварительной обработки;
- preprocessing worker - поток предварительной обработки данных;
- proxy poller - процесс опроса пассивных прокси;
- proxy group manager - менеджер балансировки нагрузки и высокой доступности прокси;
- report manager - менеджер задач формирования запланированных отчетов;
- report writer - процесс формирования запланированных отчетов;
- self-monitoring - процесс сбора внутренней статистики сервера;
- service manager - процесс управления сервисами путем получения информации о проблемах, тегах проблем и восстановлении проблем от процессов синхронизации истории, управления задачами и управления оповещениями;
- snmp poller - асинхронный процесс опроса для проверок SNMP с рабочим потоком (только элементы данных walk[OID] и get[OID]);
- snmp trapper - процесс приема SNMP-трапов;
- supervisor - координирует запуск и остановку процессов и потоков сервера, а также отслеживает их готовность;
- task manager - процесс удаленного выполнения задач, запрошенных другими компонентами (например, закрытие проблемы, подтверждение проблемы, немедленная проверка значения элемента данных, функциональность удаленных команд);
- telemetry query poller - процесс опроса для проверок элементов данных запросов телеметрии;
- timer - таймер обработки периодов обслуживания;
- trapper - процесс приема данных для активных проверок, трапов и взаимодействия с прокси;
- trigger housekeeper - процесс удаления проблем и событий, созданных триггерами, которые впоследствии были удалены;
- unreachable poller - процесс опроса недоступных устройств;
- vmware collector - сборщик данных VMware, отвечающий за получение данных от сервисов VMware.

Типы этих процессов можно наблюдать в файле журнала сервера.

Файл журнала сервера создается с разрешениями на чтение и запись только для владельца файла. Кроме того, группа владельца может читать этот файл. Все остальные разрешения запрещены.

Различные типы процессов сервера Zabbix можно отслеживать с помощью внутреннего элемента данных zabbix[process,<type>,

Статистика транзакций history syncer

Заголовок процесса history syncer отображает подробную статистику о транзакциях history syncer:

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

В "A+B triggers":

- A - триггеры, обработанные из-за значений истории;
- B - триггеры, обработанные из-за таймеров.

Временные показатели в processed...in N (<timings>) sec:

- Время, затраченное на запись значений элементов данных в базу данных;
- Время, затраченное на обновление данных элемента данных (состояние, ошибки, инвентарь узла сети и т. д.);
- Время, затраченное на запись трендов в базу данных;
- Время, затраченное на вычисление триггеров;
- Время, затраченное на обработку событий и действий.

Процедура housekeeping

Процесс housekeeper периодически удаляет устаревшие данные (историю и тренды элементов данных, пользовательские сеансы, события и т. д.), а также данные, оставшиеся после удаления объектов. Он работает циклами, при этом частота запуска и лимит удаления за цикл определяются параметрами HousekeepingFrequency и MaxHousekeeperDelete. Любые данные, не удаленные за один цикл, переносятся в следующий. Автоматический housekeeping можно включить и настроить в *Administration* > *Housekeeping*.

Когда процесс housekeeper удаляет данные, оставшиеся после удаления объектов, он использует задачи из таблицы housekeeper, которая заполняется каждый раз при удалении объекта. Например, при удалении узла сети Zabbix удаляет его элементы данных, но не их историю, тренды или проблемы. Вместо этого триггеры базы данных заполняют таблицу housekeeper задачами, состоящими из следующих полей:

- housekeeperid - ID задачи
- object - тип объекта (0 - элемент данных; 1 - триггер; 2 - service; 3 - обнаруженный узел сети; 4 - обнаруженный service)

- `objectid` - ID объекта (помогает housekeeper найти данные, связанные с объектом)

Например, при удалении узла сети с двумя элементами данных и одним триггером таблица `housekeeper` заполняется следующим образом:

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

Триггеры базы данных заполняют таблицу `housekeeper` задачами, не проверяя наличие данных, связанных с объектом; эта проверка выполняется процессом `housekeeper` при удалении данных:

- Для удаленных элементов данных (включая правила LLD) - удаляет их данные из всех таблиц истории и трендов (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Также удаляет их внутренние события из таблицы `problem`.
- Для удаленных триггеров - удаляет их события из таблиц, связанных с событиями (`problem`, `event_symptom`, `events`; сама база данных также удаляет события из таблицы `event_recovery`). Также уведомляет процесс `service manager` об удаленных событиях.

Note:
Отдельный процесс `trigger housekeeper` выполняет более узкую задачу — удаление проблем и событий, у которых нет известного исходного триггера. Частота его запуска определяется параметром `ProblemHousekeepingFrequency`.
Пока процедура housekeeping для триггеров не запущена, проблемы, созданные триггерами, которые к этому моменту уже были удалены, все еще могут генерировать проблемы сервисов и назначаться сервисам. Если в вашей конфигурации используется много правил расчета статуса сервисов, основанных на часто обнаруживаемых/необнаруживаемых триггерах, рассмотрите возможность увеличения частоты процедуры housekeeping путем изменения параметра конфигурации сервера `ProblemHousekeepingFrequency`.

- Для удаленных сервисов - удаляет их проблемы из таблицы `problem`; их события из таблицы `events` удаляются только после завершения настроенного периода хранения данных.
- Для удаленных обнаруженных узлов сети/сервисов (network discovery) - удаляет их события из таблицы `events`.

Процесс `housekeeper` удаляет только те события, которые не связаны с проблемами. Например, устаревшее событие проблемы/восстановления не будет удалено, если оно связано с открытой проблемой. Когда `housekeeper` удаляет устаревшие сущности, он сначала удаляет проблемы, а затем события.

Note:
Таблицы, использующие режим `partition` (разделенные таблицы TimescaleDB), пропускаются; обрабатываются только таблицы, использующие режим `regular`.

Поддерживаемые платформы Из-за требований к безопасности и критически важного характера работы сервера UNIX является единственной операционной системой, которая может стабильно обеспечивать необходимую производительность, отказоустойчивость и надежность. Zabbix работает на ведущих версиях рынка.

Сервер Zabbix протестирован на следующих платформах:

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

Note:
Zabbix может также работать и на других Unix-подобных операционных системах.

Локаль Обратите внимание, что сервер требует локаль UTF-8, чтобы некоторые текстовые элементы могли быть корректно интерпретированы. На большинстве современных систем, подобных Unix, локаль UTF-8 используется по умолчанию, однако на некоторых системах ее может потребоваться задать явно.

Настройка высокой доступности для серверов Zabbix

Обзор

Высокая доступность (HA) обычно требуется в критически важных инфраструктурах, которым необходимо практически полное отсутствие простоя. Таким образом, любой сервис, который потенциально может выйти из строя, должен иметь вариант аварийного переключения на случай сбоя текущего сервиса.

Zabbix предлагает **встроенное** решение высокой доступности, которое легко настроить и которое не требует предварительных глубоких знаний в этой области. Это встроенное решение может быть полезным как дополнительной уровень защиты от аппаратных или программных сбоев сервера Zabbix или для уменьшения времени простоя во время обслуживания.

В режиме высокой доступности Zabbix несколько Zabbix серверов запускаются как ноды кластера. Пока один Zabbix сервер в кластере является активной нодой, остальные находятся в режиме ожидания, готовыми перенять управление при необходимости.



Переключение на Zabbix HA ни к чему не обязывает. Вы можете переключиться обратно на автономный режим в любой момент.

Смотрите также: Детали реализации

Включение высокой доступности

Запуск сервера Zabbix в качестве узла кластера

Для запуска сервера Zabbix в качестве узла кластера в конфигурации сервера необходимо указать два параметра:

- Параметр `HANodeName` должен быть указан для каждого сервера Zabbix, который будет узлом HA-кластера.

Это уникальный идентификатор узла (например, `zabbix-node-01`), под которым сервер будет указан в конфигурациях агента и прокси. Если `HANodeName` не указан, сервер будет запущен в автономном режиме.

- Параметр `NodeAddress` должен быть указан для каждого узла.

Параметр `NodeAddress` (`address:port`) будет использоваться веб-интерфейсом Zabbix для подключения к активному узлу сервера. `NodeAddress` должен соответствовать IP-адресу или имени FQDN соответствующего сервера Zabbix.

Перезапустите все серверы Zabbix после внесения изменений в файлы конфигурации. Теперь они будут запущены как узлы кластера. Новый статус серверов можно просмотреть в разделе *Отчёты* > *Сведения о системе*, а также с помощью команды:

```
zabbix_server -R ha_status
```

Эта команда времени выполнения записывает текущий статус HA-кластера в журнал сервера Zabbix (а также выводит его в stdout):

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

Подготовка веб-интерфейса

Убедитесь, что `address:port` сервера Zabbix не указан в файле конфигурации веб-интерфейса Zabbix (`zabbix.conf.php`).

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

Веб-интерфейс Zabbix автоматически определит активный узел, прочитав настройки из таблицы узлов в базе данных Zabbix. Адрес узла активного узла будет использоваться в качестве адреса сервера Zabbix.

Конфигурация прокси

Узлы кластера HA (серверы) должны быть перечислены в конфигурации либо пассивного, либо активного прокси Zabbix.

Для пассивного прокси имена узлов должны быть перечислены в параметре Server прокси, разделенные **запятой**.

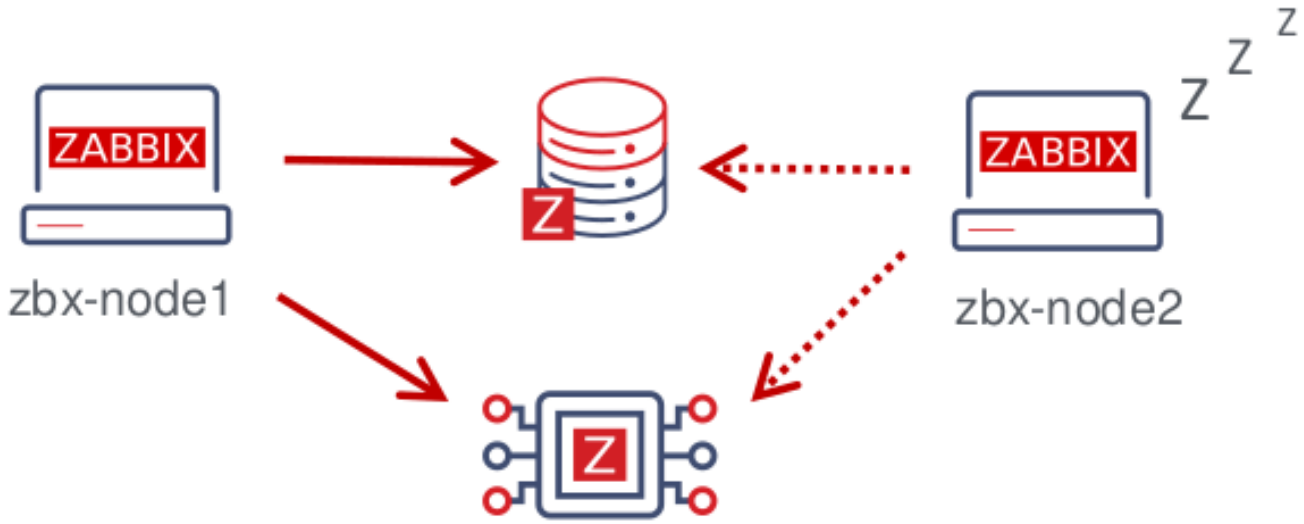
`Server=zabbix-node-01,zabbix-node-02`

Для активного прокси имена узлов должны быть перечислены в параметре Server прокси, разделенные **точкой с запятой**.

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

Конфигурация агента

Узлы кластера HA (серверы) должны быть перечислены в конфигурации Zabbix agent или Zabbix agent 2.



Чтобы включить пассивные проверки, имена узлов должны быть перечислены в параметре Server parameter, разделенные **запятой**.

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

Чтобы включить активные проверки, имена узлов должны быть перечислены в параметре ServerActive parameter. Обратите внимание, что для активных проверок узлы должны быть отделены запятой от любых других серверов, а сами узлы должны быть разделены **точкой с запятой**, например:

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

Автоматическое переключение на резервную ноду

Zabbix автоматически переключится на другую ноду, если активная нода остановится. Для автоматического переключения необходимо иметь по крайней мере одну ноду в состоянии резерва.

Как быстро произойдет автоматическое переключение? Все ноды обновляют своё время последнего доступа (и состояния, если оно меняется) каждые 5 секунд. Таким образом:

- Если активная нода выключается и успевает сообщить своё состояние как «остановлена», другая нода подхватит нагрузку в течение **5 секунд**.
- Если активная нода выключается / становится недоступной без возможности обновить своё состояние, резервные ноды будут ожидать **задержку автоматического переключения** + 5 секунд перед тем как перенять нагрузку на себя.

Задержка автоматического переключения настраивается, поддерживаемый диапазон от 10 секунд до 15 минут (по умолчанию одна минута). Чтобы изменить задержку автоматического переключения, вы можете выполнить:

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

Управление кластером высокой доступности

Текущий статус кластера высокой доступности можно контролировать с помощью специальных параметров управления во время выполнения:

- `ha_status` - записать статус кластера высокой доступности в журнал сервера Zabbix (а также вывести его в stdout);
- `ha_remove_node=target` - удалить узел высокой доступности, указанный с помощью `<target>` - имени или идентификатора узла (имя или идентификатор можно получить из вывода команды `ha_status`), например:

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

Обратите внимание, что активные узлы и узлы в режиме ожидания удалить нельзя.

- `ha_set_failover_delay=delay` - задать задержку переключения кластера высокой доступности (от 10 секунд до 15 минут; поддерживаются суффиксы единиц времени, например `30s`, `1m`).

Статус узлов можно отслеживать:

- В разделе *Отчёты* > *Сведения о системе*.
- В виджете панели мониторинга *Сведения о системе*.
- С помощью параметра управления во время выполнения `ha_status` сервера (см. выше).

Внутренний элемент данных `zabbix[cluster,discovery,nodes]` можно использовать для обнаружения узлов, поскольку он возвращает JSON с информацией об узлах высокой доступности.

Отключение высокой доступности

Чтобы отключить кластер высокой доступности:

1. Сделайте резервные копии файлов конфигурации.
2. Остановите резервные ноды.
3. Удалите параметр HANodeName на основном активном сервере.
4. Перезапустите основной сервер (он запустится в автономном режиме).

Обновление кластера высокой доступности

Чтобы выполнить обновление основной (major) версии нод высокой доступности:

1. Остановите все ноды.
2. Создайте полную резервную копию базы данных.
3. Если база данных использует репликацию, убедитесь, что все узлы синхронизированы и не имеют проблем. Не обновляйтесь, если репликация нарушена.
4. Выберите один узел, который будет выполнять обновление базы данных, измените его конфигурацию на автономный режим, закомментировав `HANodeName` и обновите его.
5. Убедитесь, что обновление базы данных полностью завершено (*Информация о системе* должна отображать, что сервер Zabbix работает).
6. Перезапустите ноду в режиме HA.
7. Обновите и запустите остальные ноды (переводить их в автономный режим не требуется, так как база данных к этому моменту уже обновлена).

При обновлении минорной версии достаточно обновить первую ноду, убедиться, что она обновлена и работает, а затем начать обновление на следующей ноде.

Детали реализации

Кластер высокой доступности (HA) является опциональным решением и поддерживается для Zabbix сервера. Собственное решение HA разработано с учётом более простого использования, это решение будет работать между разными площадками и не потребует никаких особых требований к базам данных, которые распознаются Zabbix. Пользователи могут свободно выбирать — использовать собственное решение Zabbix HA или же стороннее решение HA, в зависимости от того, какое решение наилучшим образом подходит под требования высокой доступности в их среде.

Решение состоит из нескольких экземпляров `zabbix_server` или нод. Каждая нода:

- Настраивается отдельно.
- Использует ту же базу данных.
- Имеет несколько режимов: активная, резервная, недоступная, остановленная.

Только одна нода может быть активной (рабочей) в один момент времени. На резервной ноде работает только один процесс — HA manager. Резервная нода не выполняет сбор данных, обработку или какие-либо другие обычные действия сервера; она не слушает порты; имеет минимальное количество подключений к базе данных.

Как активные, так и резервные ноды выполняют обновление времени своего последнего доступа каждые 5 секунд. Каждая резервная нода наблюдает за временем последнего доступа активной ноды. Если время последнего доступа активной ноды превышает «задержку при аварийном переключении» в секундах, резервная нода переключает себя в в режим активной ноды и назначает состояние «недоступна» ранее активной ноде.

Активная нода отслеживает своё собственное подключение к базе данных — если подключение потеряно более чем на −5 секунд, нода должна остановить всю обработку и переключиться в резервный режим. Активная нода также отслеживает состояния резервных нод — если время последнего доступа резервной ноды превышает «задержку при аварийном переключении» в секундах, резервной ноде присваивается состояние «недоступная».

Ноды разрабатывались таким образом, чтобы быть совместимыми между минорными версиями Zabbix.

Параметры конфигурации: Zabbix сервер

Обзор

Параметры конфигурации позволяют настраивать поведение сервера Zabbix.

Укажите значения параметров в файле конфигурации сервера Zabbix (`zabbix_server.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его имя)
- Подробные сведения о параметрах

Параметр	Описание
AlertScriptsPath	Укажите расположение скриптов оповещений.
AllowRoot	Разрешите серверу запускаться от имени `root`.
AllowSoftwareUpdateCheck	Разрешите получение информации об обновлениях программного обеспечения с `zabbix.com`.
AllowUnsupportedDBVersions	Разрешите серверу работать с неподдерживаемыми версиями базы данных.
BridgeAdapterURL	Задайте HTTP(S)-URL адаптера моста Zabbix.
CacheSize	Задайте размер кэша конфигурации (в байтах).
CacheUpdateFrequency	Задайте частоту обновления кэша конфигурации (в секундах).
DBHost	Укажите узел базы данных (или каталог сокета).
DBName	Укажите имя базы данных. **Обязательно**.
DBPassword	Укажите пароль базы данных.
DBPort	Укажите порт узла базы данных.
DBSchema	Укажите имя схемы базы данных. Используется для PostgreSQL.
DBSocket	Укажите путь к файлу сокета MySQL.
DBUser	Укажите пользователя базы данных.
DBTLSConnect	Укажите значение, принудительно устанавливающее TLS-соединение с базой данных.
DBTLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата базы данных.
DBTLSCertFile	Укажите полный путь к файлу, содержащему сертификат сервера Zabbix для аутентификации в базе данных.
DBTLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.
DBTLSCipher	Укажите список шифров шифрования, разрешенных сервером Zabbix для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.
DBTLSCipher13	Укажите список наборов шифров, разрешенных сервером Zabbix для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.
DebugLevel	Задайте уровень отладки.
EnableGlobalScripts	Включите глобальные скрипты на сервере Zabbix.
EnableMobileDevices	Включите поддержку мобильных устройств на сервере Zabbix.

Параметр	Описание
ExportDir	Укажите каталог для включения экспорта событий, истории и трендов в реальном времени.
ExportFileSize	Задайте максимальный размер файла экспорта (в байтах).
ExportType	Укажите типы сущностей для экспорта в реальном времени (события, история, тренды).
ExternalScripts	Укажите расположение внешних скриптов.
Fping6Location	Укажите расположение fping6.
FpingLocation	Укажите расположение fping.
FrontendAllowedIP	Укажите IP-адреса или диапазоны CIDR, которым разрешено подключаться из веб-интерфейса.
HANodeName	Укажите имя узла в кластере высокой доступности.
HistoryCacheSize	Задайте размер кэша истории.
HistoryIndexCacheSize	Задайте размер кэша индексов истории.
HistoryProvider	Задайте поставщика хранилища истории (Elasticsearch, ClickHouse или SQL).
HistoryStorageDateIndex	Этот параметр **устарел** (используйте HistoryProvider). Включите хранение значений истории в разных индексах в зависимости от даты.
HistoryStorageURL	Этот параметр **устарел** (используйте HistoryProvider). Укажите URL хранилища истории.
HistoryStorageTypes	Этот параметр **устарел** (используйте HistoryProvider). Укажите типы значений, отправляемых в хранилище истории.
HousekeepingFrequency	Задайте частоту процедуры очистки (в часах).
Include	Укажите каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
JavaGateway	Укажите IP-адрес (или имя узла сети) Java gateway Zabbix.
JavaGatewayPort	Укажите порт Java gateway Zabbix.
ListenBacklog	Задайте максимальное количество ожидающих подключений в очереди TCP.
ListenIP	Укажите IP-адрес прослушивания траппера.
ListenPort	Укажите порт прослушивания траппера.
LoadModule	Укажите модуль, который необходимо загрузить при запуске сервера.
LoadModulePath	Укажите полный путь к расположению модулей сервера.
LogFile	Укажите файл журнала сервера.
LogFileSize	Задайте максимальный размер файла журнала.
LogSlowQueries	Задайте максимальную продолжительность выполнения запроса к базе данных, после которой он будет записан в журнал (в миллисекундах).
LogType	Укажите тип вывода журнала.
MaxConcurrentChecksPerPoller	Задайте максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым опросчиком HTTP-агента, опросчиком агента или опросчиком SNMP.
MaxHousekeeperDelete	Задайте максимальное количество строк, удаляемых для одной задачи за один цикл очистки.
NodeAddress	Укажите IP-адрес или имя узла сети с необязательным портом, чтобы переопределить способ подключения веб-интерфейса к серверу.
PidFile	Укажите PID-файл.
ProblemHousekeepingFrequency	Задайте частоту удаления проблем для удаленных триггеров.
ProxyConfigFrequency	Задайте частоту отправки сервером данных конфигурации прокси Zabbix.
ProxyDataFrequency	Задайте частоту запроса сервером данных истории у прокси Zabbix.
ServiceManagerSyncFrequency	Задайте частоту синхронизации конфигурации менеджера служб.
SMSDevices	Укажите файлы модемов, которым разрешено использовать сервер Zabbix.
SNMPTrapperFile	Укажите временный файл, используемый для передачи данных от демона SNMP trap к серверу.
SocketDir	Укажите каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.
SourceIP	Укажите исходный IP-адрес.
SSHKeyLocation	Укажите расположение открытых и закрытых ключей для проверок и действий SSH.
SSLCertLocation	Укажите расположение файлов сертификатов SSL-клиента для аутентификации клиента.
SSLKeyLocation	Укажите расположение файлов закрытых ключей SSL для аутентификации клиента.
SSLCALocation	Укажите расположение файлов центра сертификации (CA) для проверки сертификата SSL-сервера.
StartAgentPollers	Задайте количество предварительно разветвленных экземпляров асинхронных опросчиков агента Zabbix.
StartAlerters	Задайте количество предварительно разветвленных экземпляров оповещателей.
StartBrowserPollers	Задайте количество предварительно разветвленных экземпляров опросчиков элементов данных браузера.
StartConnectors	Задайте количество предварительно разветвленных экземпляров рабочих процессов коннектора.

Параметр	Описание
StartDBSyncers	Задайте количество предварительно разветвленных экземпляров синхронизаторов истории.
StartDiscoverers	Задайте количество предварительно разветвленных экземпляров рабочих процессов обнаружения.
StartEscalators	Задайте количество предварительно разветвленных экземпляров эскалаторов.
StartHistoryPollers	Задайте количество предварительно разветвленных экземпляров опросчиков истории.
StartHTTPAgentPollers	Задайте количество предварительно разветвленных экземпляров асинхронных опросчиков HTTP-агента.
StartHTTPPollers	Задайте количество предварительно разветвленных экземпляров опросчиков HTTP.
StartIPMIPollers	Задайте количество предварительно разветвленных экземпляров опросчиков IPMI.
StartJavaPollers	Задайте количество предварительно разветвленных экземпляров опросчиков Java.
StartLLDProcessors	Задайте количество предварительно разветвленных экземпляров рабочих процессов низкоуровневого обнаружения (LLD).
StartODBCPollers	Задайте количество предварительно разветвленных экземпляров опросчиков ODBC.
StartPingers	Задайте количество предварительно разветвленных экземпляров пингеров ICMP.
StartPollersUnreachable	Задайте количество предварительно разветвленных экземпляров опросчиков недоступных узлов сети (включая IPMI и Java).
StartPollers	Задайте количество предварительно разветвленных экземпляров опросчиков.
StartPreprocessors	Задайте количество предварительно запущенных экземпляров рабочих процессов предварительной обработки.
StartProxyPollers	Задайте количество предварительно разветвленных экземпляров опросчиков пассивных прокси.
StartReportWriters	Задайте количество предварительно разветвленных экземпляров создателей отчетов.
StartSNMPPollers	Задайте количество предварительно разветвленных экземпляров асинхронных опросчиков SNMP.
StartSNMPTrapper	Установите значение ”1”, чтобы запустить процесс SNMP trapper.
StartTelemetryQueryPollers	Задайте количество предварительно разветвленных экземпляров асинхронных опросчиков запросов телеметрии.
StartTimers	Задайте количество предварительно разветвленных экземпляров таймеров.
StartTrappers	Задайте количество предварительно разветвленных экземпляров трапперов.
StartVMwareCollectors	Задайте количество предварительно разветвленных экземпляров сборщиков VMware.
StatsAllowedIP	Укажите IP-адреса или DNS-имена внешних экземпляров Zabbix, которым разрешено запрашивать статистику.
TelemetryProvider	Задайте источник данных телеметрии.
Timeout	Укажите, как долго (в секундах) ожидать установления соединения и обмена данными с прокси Zabbix, агентом, веб-службой и при проверках SNMP (за исключением элементов данных SNMP `walk[OID]` и `get[OID]`).
TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификатов узлов сети, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCertFile	Укажите полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCipherAll	Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределите критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Параметр	Описание
TLSCRLFile	Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного обмена данными между компонентами Zabbix.
TLSFrontendAccept	Укажите уровень шифрования для входящих подключений из веб-интерфейса.
TLSFrontendCertIssuer	Укажите разрешенного издателя сертификата веб-интерфейса.
TLSFrontendCertSubject	Укажите разрешенного субъекта сертификата веб-интерфейса.
TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ сервера, используемый для зашифрованного обмена данными между компонентами Zabbix.
TLSListen	Требуйте подключения только по TLS через сокет траппера.
TmpDir	Укажите временный каталог.
TrapperTimeout	Укажите время ожидания траппера в секундах для: - получения исторических данных от прокси Zabbix; - отправки данных конфигурации прокси Zabbix; - выполнения глобального скрипта или удаленной команды на сервере Zabbix.
TrendCacheSize	Задайте размер кэша трендов.
TrendFunctionCacheSize	Задайте размер кэша функций трендов.
UnavailableDelay	Задайте частоту проверки доступности узла сети в период недоступности.
UnreachableDelay	Задайте частоту проверки доступности узла сети в период недостижимости.
UnreachablePeriod	Задайте, через сколько секунд недостижимости считать узел сети недоступным.
User	Сбросьте привилегии до указанного существующего пользователя в системе.
ValueCacheSize	Задайте размер кэша значений истории.
Vault	Укажите поставщика хранилища.
VaultAppRoleID	Укажите идентификатор AppRole хранилища HashiCorp.
VaultAppSecretID	Укажите секретный идентификатор AppRole хранилища HashiCorp.
VaultDBPath	Укажите расположение для получения учетных данных базы данных по ключам.
VaultPrefix	Укажите пользовательский префикс для пути или запроса хранилища.
VaultTLSCertFile	Укажите имя файла сертификата SSL, используемого для аутентификации клиента.
VaultTLSKeyFile	Укажите имя файла закрытого ключа SSL, используемого для аутентификации клиента.
VaultToken	Укажите токен аутентификации хранилища HashiCorp.
VaultURL	Укажите URL сервера хранилища.
VMwareCacheSize	Задайте размер общей памяти для хранения данных VMware.
VMwareFrequency	Задайте частоту сбора данных от одной службы VMware в секундах.
VMwarePerfFrequency	Задайте частоту получения статистики счетчиков производительности от одной службы VMware в секундах.
VMwareTimeout	Задайте максимальное количество секунд, в течение которых `vmware collector` будет ожидать ответа от службы VMware.
WebDriverURL	Укажите URL интерфейса WebDriver.
WebServiceURL	Укажите URL веб-службы в формате `http[s]://host:port/report`.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для демона, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Warning:
Слишком большое количество процессов может исчерпать лимит файловых дескрипторов на процесс для менеджера предварительной обработки.

К таким процессам относятся сбор данных опросчиками, опросчиками недоступных узлов сети, опросчиками ODBC, опросчиками HTTP, опросчиками Java, пингерами, трапперами и опросчиками прокси, а также менеджер IPMI, SNMP trapper, рабочие процессы предварительной обработки и рабочие процессы обнаружения. Исчерпание лимита файловых дескрипторов приведет к **остановке** сервера Zabbix, обычно вскоре после запуска, но иногда это может занять больше времени.

Чтобы избежать подобных проблем, проверьте файл конфигурации сервера Zabbix, чтобы оптимизировать количество одновременных проверок и процессов. Кроме того, при необходимости убедитесь, что лимит файловых дескрипторов установлен достаточно высоким, проверив и скорректировав системные ограничения.

Сведения о параметре

AlertScriptsPath

Укажите расположение скрипта оповещения. Зависит от переменной установки datadir, задаваемой на этапе компиляции.

По умолчанию: /usr/local/share/zabbix/alertscripts

AllowRoot

Разрешить серверу запускаться от имени root. Если эта опция отключена и сервер запущен от root, сервер попытается переключиться на пользователя zabbix. Не имеет эффекта, если запуск выполнен от обычного пользователя.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

AllowSoftwareUpdateCheck

Разрешает веб-интерфейсу Zabbix получать информацию об обновлениях программного обеспечения с zabbix.com.

По умолчанию: 1
 Значения: 0 - не разрешать; 1 - разрешать

РазрешитьUnsupportedDBVersions

Разрешить серверу работать с неподдерживаемыми версиями базы данных.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

BridgeAdapterURL

Укажите HTTP(S)-URL адаптера моста Zabbix.

Пример:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Установите размер кэша конфигурации (в байтах). Размер общей памяти для хранения данных узлов сети, элементов данных и триггеров.

По умолчанию: 32M
 Диапазон: 128K-64G

CacheUpdateFrequency

Установите частоту обновления кэша конфигурации (в секундах). См. также параметры runtime control.

По умолчанию: 10
 Диапазон: 1-3600

DBHost

Укажите хост базы данных (или каталог сокета).

Для MySQL:

- localhost или пустая строка используют сокет UNIX-домена по умолчанию.

Для PostgreSQL:

- localhost разрешается через DNS (обычно в 127.0.0.1).
- Пустая строка использует сокет UNIX-домена по умолчанию.
- Путь (например, /var/run/pgbouncer) приводит к использованию сокета UNIX по этому пути.
- Список, разделенный запятыми, может содержать несколько значений host:port. Каждый хост проверяется по порядку, пока не будет установлено соединение с правом чтения и записи.

Пример: DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]

По умолчанию: localhost

DBName

Укажите имя базы данных.

Обязательно: **Да**

DBPassword

Укажите пароль базы данных. Закомментируйте эту строку, если пароль не используется.

DBPort

Укажите порт узла сети базы данных Zabbix (см. DBHost).
 DBSocket и DBPort взаимоисключающие параметры в конфигурации сервера. Укажите только один из них или оставьте оба не заданными.

По умолчанию для MySQL: 3306

По умолчанию для PostgreSQL: 5432

Диапазон: 1024-65535

DBSchema

Укажите имя схемы базы данных. Используется для PostgreSQL.

DBSocket

Укажите путь к файлу сокета MySQL.
 `DBSocket` и `DBPort` взаимоисключают друг друга в конфигурации сервера. Укажите только один из них или оставьте оба не заданными.

DBUser

Укажите пользователя базы данных.

DBTLSConnect

Укажите значение, чтобы принудительно использовать TLS-соединение с базой данных: - *required* - подключаться с использованием TLS; - *verify_ca* - подключаться с использованием TLS и проверять сертификат; - *verify_full* - подключаться с использованием TLS, проверять сертификат и проверять, что идентификатор базы данных, указанный в `DBHost`, соответствует ее сертификату.

Для MySQL, начиная с версии 5.7.11, и PostgreSQL поддерживаются следующие значения: `required`, `verify_ca`, `verify_full`.
 Для MariaDB, начиная с версии 10.2.6, поддерживаются значения `required` и `verify_full`.
По умолчанию не задано ни одно из значений, и поведение зависит от конфигурации базы данных.

DBTLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата базы данных.

Обязательно: нет (да, если для DBTLSConnect задано значение `verify_ca` или `verify_full`)

DBTLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат сервера Zabbix для аутентификации в базе данных.

DBTLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.

DBTLSCipher

Укажите список шифров шифрования, которые сервер Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.

DBTLSCipher13

Укажите список наборов шифров, которые сервер Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).
 См. также параметры runtime control.

По умолчанию: 3
 Диапазон: 0-5

EnableGlobalScripts

Включить глобальные скрипты на сервере Zabbix.
 Примечание: выполнение глобальных скриптов включено по умолчанию. Однако для новых установок, начиная с Zabbix 7.0, параметру EnableGlobalScripts явно присваивается значение 0 (отключено).

По умолчанию: 1
 Значения: 0 - отключить; 1 - включить

EnableMobileDevices

Включить поддержку мобильных устройств на сервере Zabbix.

Если параметр включен, сервер Zabbix будет обрабатывать запросы на добавление или удаление мобильных устройств и отправлять push-уведомления на эти устройства.

Значение по умолчанию: 1
 Значения: 0 — отключить; 1 — включить

ExportDir

Укажите каталог, чтобы включить экспорт в реальном времени событий, истории и трендов.

ExportFileSize

Установите максимальный размер файла экспорта (в байтах). Используется для ротации, если задан `ExportDir`.

По умолчанию: `1G`
 Диапазон: 1M-1G

ExportType

Укажите типы сущностей для экспорта в реальном времени (события, история, тренды). Список, разделенный запятыми. По умолчанию экспортируются все типы. Действительно только если задан `ExportDir`.
 Примечание: если `ExportType` указан, но `ExportDir` не задан, это является ошибкой конфигурации, и сервер не запустится.

Пример для экспорта истории и трендов:

```
ExportType=history,trends
```

Пример только для экспорта событий:

```
ExportType=events
```

ExternalScripts

Укажите расположение внешнего скрипта. Зависит от переменной установки `datadir`, заданной на этапе компиляции.

По умолчанию: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Укажите расположение fping6. Убедитесь, что двоичный файл fping6 принадлежит root и для него установлен флаг SUID. Оставьте пустым (`Fping6Location=`), если ваша утилита fping может обрабатывать IPv6-адреса.

По умолчанию: `/usr/sbin/fping6`

FpingLocation

Укажите расположение fping. Убедитесь, что двоичный файл fping принадлежит root и для него установлен флаг SUID.

По умолчанию: `/usr/sbin/fping`

FrontendAllowedIP

Укажите IP-адреса или диапазоны CIDR, которым разрешено подключаться из веб-интерфейса. Список, разделенный запятыми. По умолчанию для запросов веб-интерфейса принимаются все подключения. Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются эквивалентными, а `::/0` разрешит любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес.

Пример:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Укажите имя узла в кластере высокой доступности. Если значение пустое, сервер работает в автономном режиме, и создается узел с пустым именем.

HistoryCacheSize

Установите размер кэша истории (в байтах). Размер разделяемой памяти для хранения данных истории.

По умолчанию: `16M`
 Диапазон: 128K-16G

HistoryIndexCacheSize

Установите размер кэша history-index (в байтах). Размер общей памяти для индексирования данных истории, хранящихся в кэше истории. Для кэширования одного элемента данных размер кэша индекса требуется примерно 100 байт.

По умолчанию: `4M`
 Диапазон: 128K-16G

HistoryProvider

Укажите провайдер хранилища истории в формате `<provider>[;<options>]`, где:

- `<provider>` - один из поддерживаемых провайдеров хранилища истории: `sql` (по умолчанию), `elasticsearch` или `clickhouse`.
- `<options>` - пары `key=value`, разделенные запятыми. Значения могут быть заключены в кавычки.

Параметры для всех провайдеров хранилища истории:

- `value_types` - задает типы значений для хранения (`uint`, `dbl`, `str`, `log`, `text`, `json`). Каждый тип можно использовать только один раз и только для одного провайдера.
- `precache` - задает 1 (по умолчанию), чтобы предварительно загружать данные из провайдера хранилища истории в кэш значений пакетно. Это может значительно улучшить прогрев кэша значений после перезапуска.

Параметры для Elasticsearch и ClickHouse:

- `url` - задает URL HTTP или HTTPS, используемый для подключения к провайдеру хранилища истории.
- `log_slow_queries` - задает максимальную длительность (в миллисекундах), которую может занимать запрос к провайдеру хранилища истории, прежде чем он будет записан в журнал. Этот параметр становится доступным начиная с `DebugLevel=3`.
- `source_ip` - задает исходный IP-адрес, используемый для исходящих подключений к провайдеру хранилища истории.

Параметры для Elasticsearch:

- `date_index` - задает 1, чтобы включить хранение значений истории в разных индексах в зависимости от даты.

Параметры для ClickHouse:

- `db` - задает имя базы данных.
- `username` - задает имя пользователя, используемое для входа в базу данных.
- `password` - задает пароль, используемый для входа в базу данных.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - задайте эти параметры для настройки защищенного соединения (TLS/SSL) с базой данных.

Пример для ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Пример для Elasticsearch:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

Пример для ClickHouse и Elasticsearch:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Этот параметр устарел (вместо него используйте HistoryProvider).
 Включает хранение значений истории в разных индексах в зависимости от даты. Этот параметр используется при настройке Elasticsearch.

По умолчанию: 0
 Значения: 0 - отключено; 1 - включено

HistoryStorageURL

Этот параметр устарел (вместо него используйте HistoryProvider).
 Укажите URL хранилища истории. Этот параметр используется при настройке Elasticsearch.

HistoryStorageTypes

Этот параметр устарел (вместо него используйте HistoryProvider).
 Укажите типы значений, которые будут отправляться в хранилище истории. Список, разделенный запятыми. Этот параметр используется при настройке Elasticsearch.

По умолчанию: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Установите частоту выполнения процедуры очистки (в часах). Очистка удаляет устаревшую информацию из базы данных.
Примечание: Чтобы не перегружать housekeeper (например, когда периоды history и trend значительно сокращены), за один цикл очистки для каждого элемента данных удаляется не более чем в 4 раза больше часов устаревшей информации, чем значение `HousekeepingFrequency`. Таким образом, если `HousekeepingFrequency` равен 1, за один цикл будет удалено не более 4 часов устаревшей информации (начиная с самой старой записи).
Примечание: Чтобы снизить нагрузку при запуске сервера, очистка откладывается на 30 минут после старта сервера. Таким образом, если `HousekeepingFrequency` равен 1, первая процедура очистки после запуска сервера будет выполнена через 30 минут, а затем будет повторяться с задержкой в один час.
Можно отключить автоматическую очистку, установив `HousekeepingFrequency` в 0. В этом случае процедуру очистки можно запускать только с помощью параметра управления во время выполнения `housekeeper_execute`, а период устаревшей информации, удаляемой за один цикл

очистки, составляет 4 раза период с момента последнего цикла очистки, но не менее 4 часов и не более 4 дней.
См. также параметры и подробности управления во время выполнения и процедуры очистки.

По умолчанию: 1
 Диапазон: 0-24

Включение

Укажите каталог или отдельные файлы, которые нужно включить в файл конфигурации.
 Во время установки Zabbix этот параметр устанавливается в `/usr/local/etc`, если вы не измените его на этапе компиляции.

Общие правила для параметра `Include`:

- Путь может быть относительным к расположению файла `zabbix_server.conf`.
- В одном файле конфигурации можно указать несколько параметров `Include`.
- Zabbix обрабатывает параметр `Include` сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила для включения каталогов или шаблонов файлов:

- Не следует предполагать какой-либо порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общий параметр более специфичным), так как невозможно предсказать, какое значение будет использовано.
- Обратите внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан `include/my_specific.conf.BAK`), поскольку будут включены оба файла.

Чтобы включать только нужные файлы из каталога, используйте подстановочный знак (`*`) для сопоставления с шаблоном.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Укажите IP-адрес или имя хоста Zabbix Java gateway. Требуется только в том случае, если запущены Java pollers.

JavaGatewayPort

Укажите порт Zabbix Java gateway.

По умолчанию: 10052
 Диапазон: 1024-32767

ListenBacklog

Установите максимальное число ожидающих соединений в очереди TCP.
Значение по умолчанию — жестко заданная константа, которая зависит от системы.
Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть незаметно усечены до 'implementation-specified maximum'.

По умолчанию: `SOMAXCONN`
 Диапазон: 0 - INT_MAX

ListenIP

Укажите IP-адрес для прослушивания trapper. Список, разделенный запятыми.
 Trapper будет прослушивать все сетевые интерфейсы, если этот параметр не указан.

По умолчанию: `0.0.0.0`

ListenPort

Укажите порт прослушивания trapper.

По умолчанию: 10051
 Диапазон: 1024-32767

LoadModule

Укажите модуль, который необходимо загрузить при запуске сервера. Модули используются для расширения функциональности сервера. Модуль должен находиться в каталоге, указанном в `LoadModulePath`, либо перед именем модуля необходимо указать путь. Если указанный путь является абсолютным (начинается с /), параметр `LoadModulePath` игнорируется.
 Форматы:

- `LoadModule=<module.so>`
- `LoadModule=<path/module.so>`

- LoadModule=</abs_path/module.so>

Можно указать несколько параметров LoadModule.

LoadModulePath

Укажите полный путь к расположению модулей сервера. Значение по умолчанию зависит от параметров компиляции.

LogFile

Укажите файл журнала сервера.

Обязательно: Да, если LogType установлен в file; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: если достигнут предел размера файла журнала и ротация файла по какой-либо причине не удается, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024
 Обязательно: Да, если LogType установлен в file; в противном случае — нет

LogSlowQueries

Установите максимальную длительность выполнения запроса к базе данных, по истечении которой он будет записан в журнал (в миллисекундах).
0 - не записывать медленные запросы.
Этот параметр становится доступным начиная с DebugLevel=3.

По умолчанию: 0
 Диапазон: 0-3600000

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *system* - записывать журнал в syslog; - *console* - записывать журнал в стандартный вывод.

По умолчанию: file

MaxConcurrentChecksPerPoller

Задайте максимальное количество асинхронных проверок, которые могут выполняться одновременно каждым опрашивателем HTTP-агента, опрашивателем агента или опрашивателем SNMP. См. StartHTTPAgentPollers, StartAgentPollers и StartSNMPPollers.

По умолчанию: 1000
 Диапазон: 1-1000

MaxHousekeeperDelete

Установите максимальное количество строк, удаляемых за одну задачу в одном цикле housekeeping из таблиц history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint и problem.
Задачи housekeeping существуют в таблице housekeeper в формате: [housekeeperid], [object], [objectid].
 Этот параметр применяется только к удалению данных, оставшихся после уже удаленных элементов данных.
 Если установлено значение 0, ограничение не используется вовсе.
 В этом случае вы должны точно понимать, что делаете, чтобы не перегрузить базу данных.
 Когда удаляется много элементов данных, это увеличивает нагрузку на базу данных, потому что housekeeper должен удалить все исторические данные, которые были у этих элементов.
 Например, если нужно удалить только 1 прототип элемента данных из шаблона, но этот шаблон связан с 50 узлами сети и для каждого узла прототип разворачивается в 100 реальных элементов данных, в общей сложности нужно удалить 5000 элементов данных (1*50*100).
 Если для MaxHousekeeperDelete задано значение 500 (MaxHousekeeperDelete=500), процесс housekeeper должен будет удалить до 2500000 значений (5000*500) для удаленных элементов данных из таблиц history и trends за один цикл.

См. также подробности в процедуре housekeeping.

По умолчанию: 5000
 Диапазон: 0-1000000

NodeAddress

Укажите IP-адрес или имя хоста с необязательным портом, чтобы переопределить способ подключения веб-интерфейса к серверу.
Формат: <address>[:<port>]

Если IP-адрес или имя хоста не заданы, будет использовано значение ListenIP. Если ListenIP не задан, будет использовано значение localhost.
Если порт не задан, будет использовано значение ListenPort. Если ListenPort не задан, будет использовано значение 10051.

Этот параметр можно переопределить адресом, указанным в конфигурации веб-интерфейса.

См. также: параметр HANodeName; Включение высокой доступности.

По умолчанию: localhost:10051

PidFile

Укажите PID-файл.

По умолчанию: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Установите частоту удаления проблем для удаленных триггеров (в секундах).
 См. также подробности в процедуре очистки.

По умолчанию: 60
 Диапазон: 1-3600

ProxyConfigFrequency

Установите частоту отправки сервером конфигурационных данных на прокси Zabbix. Используется только для прокси в пассивном режиме.

По умолчанию: 10
 Диапазон: 1-604800

ProxyDataFrequency

Задайте частоту запроса сервером исторических данных у прокси Zabbix. Используется только для прокси в пассивном режиме.

По умолчанию: 1
 Диапазон: 1-3600

ServiceManagerSyncFrequency

Установите частоту синхронизации конфигурации менеджера служб (в секундах).

По умолчанию: 60
 Диапазон: 1-3600

SMSDevices

Укажите файлы модемов, которым разрешено использовать сервер Zabbix. Список, разделенный запятыми.
 Отправка SMS невозможна, если этот параметр не задан.

Пример:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Укажите временный файл, используемый для передачи данных от демона SNMP trap к серверу.
Должен совпадать с параметром в `zabbix_trap_receiver.pl` или в файле конфигурации SNMPTT.

По умолчанию: `/tmp/zabbix_traps.tmp`

SocketDir

Укажите каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.

По умолчанию: `/tmp`

SourceIP

Укажите исходящий IP-адрес для:

- исходящих подключений к прокси Zabbix и агенту Zabbix
- подключений без использования агента (VMware, SSH, JMX, SNMP, Telnet и простых проверок)
- подключений HTTP-агента
- HTTP-запросов JavaScript для элемента данных типа script
- HTTP-запросов JavaScript на этапе предварительной обработки
- отправки уведомлений по электронной почте (подключения к SMTP-серверу)
- уведомлений вебхуков (HTTP-подключения JavaScript)
- подключений к Vault

SSHKeyLocation

Укажите расположение открытых и закрытых ключей для SSH-проверок и действий.

SSLCertLocation

Укажите расположение файлов сертификатов SSL-клиента для аутентификации клиента.
 Этот параметр используется в веб-мониторинге, для связи с Vault, для хранилища истории ClickHouse и источника телеметрических данных ClickHouse.

SSLKeyLocation

Укажите расположение файлов закрытых ключей SSL для аутентификации клиента.
 Этот параметр используется в веб-мониторинге, для связи с Vault, для провайдера хранения истории ClickHouse и для источника телеметрических данных ClickHouse.

SSLCALocation

Укажите расположение файлов центра сертификации (CA) для проверки сертификатов SSL-сервера. Если параметр не задан, будет использоваться общесистемный каталог.
 Обратите внимание, что значение этого параметра будет задано для параметра libcurl `CURLOPT_CAPATH`. В версиях libcurl до 7.42.0 этот параметр действует только в том случае, если libcurl был скомпилирован для использования OpenSSL. Дополнительную информацию см. на веб-странице cURL.
 Этот параметр используется в веб-мониторинге, аутентификации SMTP, элементах данных HTTP agent, для обмена данными с Vault, для провайдера хранения истории ClickHouse и для источника телеметрических данных ClickHouse.

Запуск опросчиков агента

Задайте количество предварительно созданных экземпляров Zabbix agent pollers. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartAlerters

Задает количество предварительно порожденных экземпляров alerters.

По умолчанию: 3
 Диапазон: 1-100

StartBrowserPollers

Установите количество предварительно созданных экземпляров pollers для элемента данных браузера.

По умолчанию: 1
 Диапазон: 0-1000

StartConnectors

Установите количество предварительно порожденных экземпляров connector workers. Процесс менеджера connector автоматически запускается при старте connector worker.

По умолчанию: 0
 Диапазон: 0-1000

StartDBSyncers

Установите количество предварительно порожденных экземпляров history syncers.
Примечание: Будьте осторожны при изменении этого значения, так как его увеличение может принести больше вреда, чем пользы. Примерно значение по умолчанию должно быть достаточным для обработки до 4000 NVPS.

По умолчанию: 4
 Диапазон: 1-100

StartDiscoverers

Установите количество предварительно порожденных экземпляров рабочих процессов обнаружения[1].

По умолчанию: 5
 Диапазон: 0-1000

StartEscalators

Установите количество предварительно созданных экземпляров эскалаторов.

По умолчанию: 1
 Диапазон: 1-100

StartHistoryPollers

Задает количество предварительно порожденных экземпляров history pollers.
Требуется только для вычисляемых проверок.

По умолчанию: 5
 Диапазон: 0-1000

StartHTTPAgentPollers

Задайте количество предварительно порожденных экземпляров опросчиков HTTP-агента. См. MaxConcurrentChecksPer-Poller.

По умолчанию: 1
 Диапазон: 0-1000

StartHTTPPollers

Установите количество предварительно созданных экземпляров HTTP pollers[1].

По умолчанию: 1
 Диапазон: 0-1000

StartIPMIPollers

Установите количество предварительно порожденных экземпляров IPMI pollers.

По умолчанию: 0
 Диапазон: 0-1000

StartJavaPollers

Установите количество предварительно созданных экземпляров Java pollers[1].

По умолчанию: 0
 Диапазон: 0-1000

StartLLDProcessors

Установите количество предварительно порожденных экземпляров workers низкоуровневого обнаружения (LLD)[1].
 Процесс LLD manager автоматически запускается при запуске LLD worker.

По умолчанию: 2
 Диапазон: 1-100

StartODBCPollers

Установите количество предварительно порожденных экземпляров ODBC pollers[1].

По умолчанию: 1
 Диапазон: 0-1000

StartPingers

Установите количество предварительно порожденных экземпляров ICMP pingers[1].

По умолчанию: 1
 Диапазон: 0-1000

StartPollersUnreachable

Установите количество предварительно созданных экземпляров pollers for unreachable hosts (включая IPMI и Java)[1].
 Если запущены обычные pollers, IPMI или Java pollers, должен быть запущен как минимум один poller for unreachable hosts.

По умолчанию: 1
 Диапазон: 0-1000

StartPollers

Установите количество предварительно созданных экземпляров pollers[1].

По умолчанию: 5
 Диапазон: 0-1000

StartPreprocessors

Установите количество предварительно запущенных экземпляров потоков предварительной обработки worker[1] не меньше количества доступных ядер CPU. Следует задать больше worker, если предварительная обработка не ограничена CPU и выполняет много сетевых запросов.

По умолчанию: 16
 Диапазон: 1-1000

StartProxyPollers

Установите количество предварительно созданных экземпляров pollers for passive proxies[1].

По умолчанию: 1
 Диапазон: 0-250

StartReportWriters

Установите количество предварительно порожденных экземпляров report writers.
Если установлено значение 0, плановое создание отчетов отключается.
Процесс менеджера отчетов автоматически запускается при запуске report writer.

По умолчанию: 0
 Диапазон: 0-100

StartSNMPPollers

Задает количество предварительно порожденных экземпляров SNMP pollers. См. `MaxConcurrentChecksPerPoller`.

По умолчанию: 1
 Диапазон: 0-1000

StartSNMPTrapper

Установите значение 1, чтобы запустить процесс SNMP trapper.

По умолчанию: 0
 Диапазон: 0-1

StartTelemetryQueryPollers

Задайте количество предварительно запущенных экземпляров асинхронных опросчиков запросов Telemetry. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartTimers

Установите количество предварительно созданных экземпляров timers.
Процессы timers обрабатывают периоды обслуживания.

По умолчанию: 1
 Диапазон: 1-1000

StartTrappers

Установите количество предварительно порожденных экземпляров trappers[1].
 Trappers принимают входящие соединения от Zabbix sender, активных агентов и активных прокси.

По умолчанию: 5
 Диапазон: 0-1000

StartVMwareCollectors

Задайте количество предварительно порожденных экземпляров сборщика VMware.

По умолчанию: 0
 Диапазон: 0-250

StatsAllowedIP

Укажите IP-адреса (при необходимости в нотации CIDR) или DNS-имена внешних экземпляров Zabbix, которым разрешено запрашивать статистику. Список, разделенный запятыми.
 Если этот параметр не задан, запросы статистики приниматься не будут.
 Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются эквивалентными, а `::/0` разрешит любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес.

Пример:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Задайте источник данных телеметрии в формате `<provider>[;<options>]`, где:

- `<provider>` - один из поддерживаемых источников данных телеметрии: `clickhouse`
- `<options>` - разделённые запятыми пары `key=value`. Значения можно заключать в кавычки.

Параметры для ClickHouse:

- `url` - задаёт URL HTTP или HTTPS, используемый для подключения к источнику данных телеметрии.
- `db` - задаёт имя базы данных.
- `username` - задаёт имя пользователя, используемое для входа в базу данных.
- `password` - задаёт пароль, используемый для входа в базу данных.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - задайте эти параметры для настройки защищённого (TLS/SSL) подключения к базе данных.
- `source_ip` - задаёт исходный IP-адрес, используемый для исходящих подключений к источнику данных телеметрии.
- `vault_path` - задаёт путь к секрету Vault, содержащему значения `username` и `password`. Не используйте вместе с параметрами `username` и `password`.

Пример для ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Пример для ClickHouse с Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Укажите, сколько времени ожидать (в секундах) установления соединения и обмена данными с прокси Zabbix, агентом, веб-службой и устаревшими проверками SNMP (одиночный номер OID или строка).

Этот параметр определяет длительность для различных операций связи:

- выполнение удаленной команды на агенте Zabbix
- выполнение команды SSH/Telnet
- запросы к веб-службе Zabbix
- тайм-аут связи для тестового запроса mediatype и метода `history.push`

- переназначение элементов данных, когда интерфейс IPMI становится недоступным
- отправка ответа прокси Zabbix при сбое обмена данными из-за проблем с правами доступа или шифрованием
- крайний срок для асинхронных сокетов IPC и параметров управления во время выполнения
- подключения JMX
- получение статистики с удаленного прокси Zabbix или сервера
- отправка ответов веб-интерфейсу Zabbix
- DNS-запросы асинхронного poller
- ответ на heartbeat активной проверки
- получение данных агента Zabbix (значений) от активных агентов
- получение данных от Zabbix sender
- ответ, когда trapper не может обработать запрос
- отправка списка активных проверок агенту Zabbix

Этот тайм-аут **не** будет использоваться для тех проверок, для которых в веб-интерфейсе настроены параметры гибкого тайм-аута (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных). Например, элементы данных SNMP `walk[OID]` и `get[OID]` используют тайм-аут, настроенный в веб-интерфейсе; устаревшие проверки SNMP по-прежнему используют значение тайм-аута сервера.

По умолчанию: 3
 Диапазон: 1-30

TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата узла-пира, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Укажите полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемых для зашифрованной связи между компонентами Zabbix.

TLSCipherAll

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

TLSCipherPSK

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSFrontendAccept

Укажите уровень шифрования для входящих соединений от веб-интерфейса.

Можно указать несколько значений, разделяя их запятыми:

- unencrypted - принимать соединения без шифрования.
- cert - принимать соединения, защищенные с помощью TLS и сертификата.

По умолчанию: `unencrypted`

TLSFrontendCertIssuer

Укажите разрешенного издателя сертификата веб-интерфейса.

TLSFrontendCertSubject

Укажите допустимый subject сертификата веб-интерфейса.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ сервера, используемый для зашифрованной связи между компонентами Zabbix.

TLSListen

Требовать подключения только по TLS на сокете trapper.

Поддерживаемые значения:

- required - принимать только подключения TLS

TmpDir

Укажите временный каталог.

По умолчанию: `/tmp`

TrapperTimeout

Укажите тайм-аут trapper в секундах для:

- получения исторических данных от прокси Zabbix
- отправки данных конфигурации на прокси Zabbix
- выполнения глобального скрипта или удаленной команды на сервере Zabbix

По умолчанию: 300
 Диапазон: 1-300

TrendCacheSize

Задайте размер кэша трендов в байтах.
Размер разделяемой памяти для хранения трендов.

По умолчанию: `4M`
 Диапазон: 128K-16G

TrendFunctionCacheSize

Установите размер кэша функций трендов в байтах.
Размер разделяемой памяти для кэширования вычисленных данных функций трендов.

По умолчанию: `4M`
 Диапазон: 128K-2G

UnavailableDelay

Установите частоту проверки доступности узла сети в течение периода недоступности в секундах.

По умолчанию: 60
 Диапазон: 1-3600

UnreachableDelay

Задайте частоту проверки доступности узла сети в течение периода недоступности в секундах.

По умолчанию: 15
 Диапазон: 1-3600

UnreachablePeriod

Указывает, через сколько секунд недоступности считать узел сети недоступным.

По умолчанию: 45
 Диапазон: 1-3600

Пользователь

Снижает привилегии до указанного существующего пользователя в системе.
Действует только при запуске от имени 'root' и если AllowRoot отключен.

По умолчанию: `zabbix`

ValueCacheSize

Задайте размер кэша значений истории (в байтах).
 Размер разделяемой памяти для кэширования запросов данных истории элементов данных.
 Значение 0 отключает кэш значений (не рекомендуется).
 Когда в кэше значений заканчивается разделяемая память, предупреждение записывается в журнал сервера каждые 5 минут.

По умолчанию: `8M`
 Диапазон: 0, 128K-64G

Vault

Укажите поставщика хранилища: - *HashiCorp* - HashiCorp KV Secrets Engine version 2; - *CyberArk* - CyberArk Central Credential Provider
Должен соответствовать поставщику хранилища, заданному во веб-интерфейсе.

По умолчанию: `HashiCorp`

VaultAppRoleID

Укажите AppRole ID для HashiCorp Vault.
 Если задано, включает аутентификацию AppRole. Не может использоваться вместе с VaultToken.
 Поддерживается только для HashiCorp Vault.

VaultAppSecretID

Укажите secret ID AppRole в HashiCorp Vault.
 Поддерживается только для HashiCorp Vault.

VaultDBPath

Укажите расположение для получения учетных данных базы данных по ключам. Это путь или запрос Vault, в зависимости от используемого Vault.

Ключи, используемые для HashiCorp, — `password` и `username`.

Пример пути с `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Пример пути без VaultPrefix:

```
secret/zabbix/database
```

Ключи, используемые для CyberArk, — `Content` и `UserName`.

Пример:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Этот параметр можно использовать только если `DBUser` и `DBPassword` не указаны.

VaultPrefix

Укажите пользовательский префикс для пути или запроса Vault в зависимости от используемого Vault. Если значение не указано, будут использованы наиболее подходящие значения по умолчанию.
 Обратите внимание, что `data` автоматически добавляется после точки монтирования для HashiCorp, если `VaultPrefix` не указан.

Пример префикса для HashiCorp:

```
/v1/secret/data/zabbix/
```

Пример префикса для Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Укажите имя файла SSL-сертификата, используемого для аутентификации клиента. Файл сертификата должен быть в формате PEM1.
 Если файл сертификата также содержит закрытый ключ, оставьте поле файла SSL-ключа пустым.
 Каталог, содержащий этот файл, задается параметром конфигурации SSLCertLocation.
 Этот параметр можно не указывать, но для хранилища CyberArkCCP его использование рекомендуется.

VaultTLSKeyFile

Укажите имя файла закрытого ключа SSL, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM1.
Каталог, содержащий этот файл, задается параметром конфигурации SSLKeyLocation.
Этот параметр можно не указывать, но для хранилища CyberArkCCP его рекомендуется задать.

VaultToken

Укажите токен аутентификации HashiCorp Vault. Токен аутентификации HashiCorp Vault должен быть сгенерирован исключительно для сервера Zabbix с правом только на чтение для путей, указанных в макросах Vault, и правом только на чтение для пути, указанного в необязательном параметре конфигурации VaultDBPath.
 Это ошибка, если `VaultToken` и переменная среды `VAULT_TOKEN` определены одновременно.
 Не может использоваться с VaultAppRoleID.
 Поддерживается только для HashiCorp Vault.

VaultURL

Укажите URL сервера vault. Будет использоваться системный каталог сертификатов CA, если не указан SSLCALocation.

По умолчанию: `https://127.0.0.1:8200`

VMwareCacheSize

Задайте размер разделяемой памяти для хранения данных VMware. Для мониторинга использования кэша VMware можно использовать внутреннюю проверку `zabbix[vmware,buffer,...]` (см. Внутренние проверки).
Обратите внимание, что разделяемая память не выделяется, если не настроены экземпляры `vmware collector` для запуска.

По умолчанию: 8M
 Диапазон: 256K-2G

VMwareFrequency

Установите частоту сбора данных с одного VMware-сервиса (в секундах). Эта частота должна быть равна наименьшему интервалу обновления любого элемента данных мониторинга VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwarePerfFrequency

Установите частоту получения статистики счетчиков производительности от одного VMware service (в секундах).
Эта частота должна быть установлена равной наименьшему интервалу обновления любого элемента данных VMware monitoring item, который использует счетчики производительности VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwareTimeout

Установите максимальное количество секунд, в течение которых `vmware collector` будет ожидать ответа от службы VMware (vCenter или гипервизора ESX).

По умолчанию: 10
 Диапазон: 1-300

WebDriverURL

Укажите URL интерфейса WebDriver.

Пример (используется с автономным сервером Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Укажите URL веб-службы в формате `http[s]://host:port/report`.

Пример:

```
WebServiceURL=http://localhost:10053/report
```

Схему `http://` можно опустить только для соединений без TLS (HTTP); при настроенном TLS необходимо использовать `https://`.

Сноски

[1] Обратите внимание, что слишком большое количество процессов сбора данных (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) вместе с IPMI manager, SNMP trapper, preprocessing workers и discovery workers может исчерпать лимит дескрипторов файлов на процесс для preprocessing manager.

> **Warning:**
> Исчерпание лимита дескрипторов файлов приведет к остановке сервера Zabbix, обычно вскоре после запуска, но иногда это может занять больше времени. Чтобы избежать таких проблем, просмотрите файл конфигурации сервера Zabbix, чтобы оптимизировать количество одновременных проверок и процессов. Кроме того, при необходимости убедитесь, что лимит дескрипторов файлов установлен достаточно высоким, проверив и скорректировав системные ограничения.

Прокси

Обзор Прокси Zabbix — это процесс, который может собирать данные мониторинга с одного или нескольких контролируемых устройств и отправлять эту информацию серверу Zabbix, фактически действуя от имени сервера. Все собранные данные буферизуются локально, а затем передаются серверу Zabbix, которому принадлежит прокси.

Развертывание прокси необязательно, но может значительно помочь распределить нагрузку одного сервера Zabbix. Если данные собирают только прокси, обработка на сервере требует меньше ресурсов ЦП и операций ввода-вывода диска.

Прокси Zabbix — идеальное решение для централизованного мониторинга удаленных площадок, филиалов и сетей, в которых нет локальных администраторов.

Для прокси Zabbix требуется отдельная база данных.

> **Attention:**
> Обратите внимание, что прокси Zabbix поддерживает базы данных SQLite, MySQL и PostgreSQL.

См. также:

- Распределенный мониторинг с использованием прокси
- Балансировка нагрузки и высокая доступность прокси
- Синхронизация конфигурации мониторинга
- Протокол обмена данными между сервером и прокси
- Параметры конфигурации: прокси Zabbix
- Настройка облачного прокси Zabbix

Запущенный прокси Если установлен из пакета

Zabbix прокси работает как демон. Его можно запустить, выполнив:

```
systemctl start zabbix-proxy
```

Эта команда будет работать на большинстве систем GNU/Linux. На других системах вам, возможно, потребуется выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния Zabbix прокси, используйте следующие команды:

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

Запуск вручную

Если указанный выше способ не работает, необходимо запустить его вручную. Найдите путь к двоичному файлу `zabbix_proxy` и выполните:

```
zabbix_proxy
```

С Zabbix proxy можно использовать следующие параметры командной строки:

Parameter	Description
`-c --config <config-file>`	Путь к файлу конфигурации.
`-f --foreground`	Запустить Zabbix proxy в режиме переднего плана.
`-R --runtime-control <option>`	Выполнить административные функции.
`-T --test-config`	Проверить файл конфигурации и выйти.
`-h --help`	Вывести справочную информацию и выйти.
`-V --version`	Показать номер версии.

Примеры запуска Zabbix proxy с параметрами командной строки:

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

Управление во время выполнения

Параметры управления во время выполнения:

Option	Description	Target
`config_cache_reload`	Перезагрузить кэш конфигурации. Игнорируется, если кэш в данный момент загружается. Активный прокси Zabbix подключится к серверу Zabbix и запросит данные конфигурации. Пассивный прокси Zabbix запросит данные конфигурации у сервера Zabbix при следующем подключении сервера к прокси.	
`history_cache_clear[=<target>]`	Очистить кэш истории для элемента данных, указанного по его ID. Затрагивает все значения элемента данных, кроме первого и последнего значения.	**target** - ID элемента данных.
`diaginfo[=<section>]`	Собрать диагностическую информацию в файле журнала прокси.	`historycache` - статистика кэша истории; `preprocessing` - статистика менеджера предварительной обработки; `locks` - список мьютексов (пусто в системах *BSD*).
`snmp_cache_reload`	Перезагрузить кэш SNMP - очистить свойства SNMP engine (engine time, engine boots, engine id, credentials) для всех узлов сети. Используется для принудительной глобальной очистки кэша при устранении неполадок SNMP.	
`housekeeper_execute`	Запустить процедуру housekeeping. Игнорируется, если процедура housekeeping уже выполняется.	
`log_level_increase[=<target>]`	Увеличить уровень журналирования, влияет на все процессы, если target не указан. Не поддерживается в системах *BSD*.	**process type** - все процессы указанного типа (например, `poller`). См. все типы процессов прокси. **process type,N** - тип процесса и номер (например, `poller,3`) **pid** - идентификатор процесса (1 to 65535). Для больших значений укажите target как 'process type,N'.
`log_level_decrease[=<target>]`	Уменьшить уровень журналирования, влияет на все процессы, если target не указан. Не поддерживается в системах *BSD*.	

Option	Description	Target
`prof_enable[=<target>]`	Включить профилирование. Влияет на все процессы, если target не указан. Включенное профилирование предоставляет сведения обо всех rwlocks/mutexes по имени функции.	**process type** - все процессы указанного типа (например, `history syncer`). См. все типы процессов прокси. **process type,N** - тип процесса и номер (например, `history syncer,1`). **pid** - идентификатор процесса (1 to 65535). Для больших значений укажите target как 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` можно использовать с типом процесса и номером (например, `history syncer,1,processing`) или со всеми процессами типа (например, `history syncer,rwlock`).
`prof_disable[=<target>]`	Отключить профилирование. Влияет на все процессы, если target не указан.	**process type** - все процессы указанного типа (например, `history syncer`). См. все типы процессов прокси. **process type,N** - тип процесса и номер (например, `history syncer,1`). **pid** - идентификатор процесса (1 to 65535). Для больших значений укажите target как 'process type,N'.

Когда тип процесса указан как target для управления во время выполнения, команда применяется только к запущенным процессам этого типа. Если в данный момент не запущен ни один процесс указанного типа, команда возвращает ошибку:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

Пример использования управления во время выполнения для перезагрузки кэша конфигурации прокси:

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

Пример использования управления во время выполнения для очистки кэша истории для элемента данных:

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

Примеры использования управления во время выполнения для сбора диагностической информации:

```
### :
zabbix_proxy -R diaginfo

### :
zabbix_proxy -R diaginfo=historycache
```

Пример использования управления во время выполнения для перезагрузки кэша SNMP:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Когда интерфейс SNMPv3 обновляется через веб-интерфейс Zabbix, Zabbix в большинстве случаев автоматически перезагружает новые учетные данные SNMPv3 для этого интерфейса; используйте `-R snmp_cache_reload` только если опрос по-прежнему не выполняется после изменения учетных данных (например, из-за несоответствий engineBoots/engineID или устройств, не соответствующих RFC), либо когда требуется принудительно очистить глобальный кэш SNMP для устранения неполадок.

Пример использования управления во время выполнения для запуска housekeeper:

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

Примеры использования управления во время выполнения для изменения уровня журналирования:

```
### :
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### poller:
```

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

###                            PID 1234:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

###                              http poller:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Запуск от имени пользователя

Прокси Zabbix предназначен для работы от имени пользователя, не являющегося пользователем root. Он будет работать от имени любого пользователя, не являющегося пользователем root, от имени которого он был запущен. Таким образом, вы можете запускать прокси от имени любого пользователя, не являющегося пользователем root, без каких-либо проблем.

Если вы попытаетесь запустить его от имени пользователя «root», он переключится на жестко заданного пользователя «zabbix», который должен присутствовать в вашей системе. Вы можете запускать прокси от имени пользователя «root» только в том случае, если соответствующим образом измените параметр `AllowRoot` в файле конфигурации прокси.

Файл конфигурации

Путь по умолчанию к файлу конфигурации зависит от операционной системы и способа установки Zabbix:

- Для установки из исходных кодов: `/usr/local/etc/zabbix_proxy.conf` или путь, заданный переменными время компиляции `--sysconfdir` или `--prefix`.
- Для установки из пакетов: `/etc/zabbix/zabbix_proxy.conf`.

См. параметры файла конфигурации для получения подробной информации о настройке `zabbix_proxy`.

Типы процессов и потоков прокси

- `agent poller` - асинхронный процесс опроса для пассивных проверок с рабочим потоком;
- `apm manager` - процесс управления мониторингом производительности приложений (APM);
- `apm worker` - процесс обработки задач мониторинга производительности приложений (APM) от `apm manager`;
- `availability manager` - процесс обновления доступности узлов сети;
- `browser poller` - процесс опроса для проверок элементов данных браузера;
- `configuration syncer` - процесс управления кэшем данных конфигурации в памяти;
- `data sender` - процесс отправки данных прокси;
- `discovery manager` - процесс управления обнаружением устройств;
- `discovery worker` - процесс обработки задач обнаружения от discovery manager;
- `history syncer` - процесс записи данных в базу данных истории;
- `housekeeper` - процесс удаления устаревшей истории элементов данных;
- `http agent poller` - асинхронный процесс опроса для HTTP-проверок с рабочим потоком;
- `http poller` - процесс опроса веб-мониторинга;
- `icmp pinger` - процесс опроса для проверок icmpping;
- `internal poller` - процесс опроса для внутренних проверок;
- `ipmi manager` - процесс управления опросом IPMI;
- `ipmi poller` - процесс опроса для проверок IPMI;
- `java poller` - процесс опроса для проверок Java;
- `odbc poller` - процесс опроса для проверок ODBC;
- `poller` - обычный процесс опроса для пассивных проверок;
- `preprocessing manager` - процесс управления задачами предварительной обработки с рабочими потоками предварительной обработки;
- `preprocessing worker` - поток предварительной обработки данных;
- `self-monitoring` - процесс сбора внутренней статистики сервера;
- `snmp poller` - асинхронный процесс опроса для проверок SNMP с рабочим потоком (только элементы данных `walk[OID]` и `get[OID]`);
- `snmp trapper` - процесс обработки SNMP-трапов;
- `supervisor` - координирует запуск и остановку процессов и потоков сервера, а также отслеживает их готовность;
- `task manager` - процесс удаленного выполнения задач, запрошенных другими компонентами (например, закрытие проблемы, подтверждение проблемы, немедленная проверка значения элемента данных, выполнение удаленной команды);
- `telemetry query poller` - процесс опроса для проверок элементов данных запросов телеметрии;
- `trapper` - процесс обработки данных активных проверок, трапов и взаимодействия с прокси;
- `unreachable poller` - процесс опроса недоступных устройств;
- `vmware collector` - сборщик данных VMware, отвечающий за получение данных от служб VMware.

Для наблюдения за этими типами процессов можно использовать файл журнала прокси.

Файл журнала прокси создается с разрешениями на чтение и запись только для владельца файла. Кроме того, группа владельца может читать этот файл. Все остальные разрешения запрещены.

Различные типы процессов прокси можно отслеживать с помощью внутреннего элемента данных `zabbix[process,<type>,<mode>`

Статистика транзакций history syncer

Заголовок процесса history syncer отображает подробную статистику по транзакциям history syncer.

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

Временные значения в `processed...in N (<timings>) sec` означают:

- Время, затраченное на запись значений элемента данных в базу данных.
- Время, затраченное на обновление данных элемента данных (состояние, ошибки).

Процедура очистки

У прокси Zabbix есть процесс `housekeeper`, который удаляет устаревшую историю элементов данных и тренды. Он выполняется циклами, а частота определяется параметром `HousekeepingFrequency`, а лимит удаления за один цикл — параметрами `ProxyLocalBuffer` и `ProxyOfflineBuffer`. В отличие от процедуры очистки на сервере Zabbix, процесс `housekeeper` у прокси не использует таблицу `housekeeper` — он удаляет все устаревшие данные один раз за каждый цикл очистки.

Поддерживаемые платформы Прокси Zabbix работает на том же списке поддерживаемых платформ, что и сервер Zabbix.

Буфер памяти Буфер памяти позволяет сохранять новые данные (значения элементов данных, сетевое обнаружение, автоматическую регистрацию узлов сети) в буфере и загружать их на сервер Zabbix без обращения к базе данных. Буфер памяти был добавлен для прокси в Zabbix 7.0.

В установках до Zabbix 7.0 собранные данные сохранялись в базе данных перед загрузкой на сервер Zabbix. Для таких установок это остается поведением по умолчанию после обновления до Zabbix 7.0.

Для оптимизации производительности рекомендуется настроить использование буфера памяти на прокси. Это можно сделать, изменив значение `ProxyBufferMode` с `disk` (жестко заданное значение по умолчанию для существующих установок) на `hybrid` (рекомендуется) или `memory`. Также необходимо задать размер буфера памяти (параметр `ProxyMemoryBufferSize`).

В гибридном режиме буфер защищен от потери данных: если прокси остановлен, буфер заполнен или данные слишком старые, неотправленные данные записываются в базу данных. После записи всех значений в базу данных прокси снова начинает использовать буфер памяти.

В режиме памяти используется буфер памяти, однако защита от потери данных отсутствует. Если прокси остановлен или память переполнена, неотправленные данные будут удалены.

Гибридный режим (`ProxyBufferMode=hybrid`) применяется ко всем новым установкам начиная с Zabbix 7.0.

Дополнительные параметры, такие как `ProxyMemoryBufferSize` и `ProxyMemoryBufferAge`, определяют размер буфера памяти и максимальный возраст данных в буфере соответственно.

Обратите внимание, что при конфликтующей конфигурации прокси выведет ошибку и не запустится, например, если:

- для `ProxyBufferMode` задано значение `hybrid` или `memory`, а `ProxyMemoryBufferSize` имеет значение 0;
- для `ProxyBufferMode` задано значение `hybrid` или `memory`, а `ProxyLocalBuffer` имеет значение, отличное от 0.

Локаль Обратите внимание, что прокси требует локаль UTF-8, чтобы некоторые текстовые элементы могли корректно интерпретироваться. На большинстве современных Unix-подобных систем локаль UTF-8 используется по умолчанию, однако на некоторых системах ее может потребоваться задать явно.

Расчет очередей во время обслуживания

> **Attention:**
> Прокси Zabbix не знает о периодах обслуживания; подробности см. в разд
> обслуживания.

Использование веб-интерфейса Zabbix за прокси Если веб-интерфейс Zabbix работает за прокси-сервером, путь к cookie в файле конфигурации прокси необходимо переписать, чтобы он соответствовал пути, используемому обратным прокси-сервером. См. примеры ниже. Если путь к cookie не переписан, у пользователей могут возникнуть проблемы с авторизацией при попытке войти в веб-интерфейс Zabbix.

Пример конфигурации для NGINX

```
### ..
location / {
### ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
### ..
```

Пример конфигурации для Apache

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

Распределенный мониторинг с использованием прокси

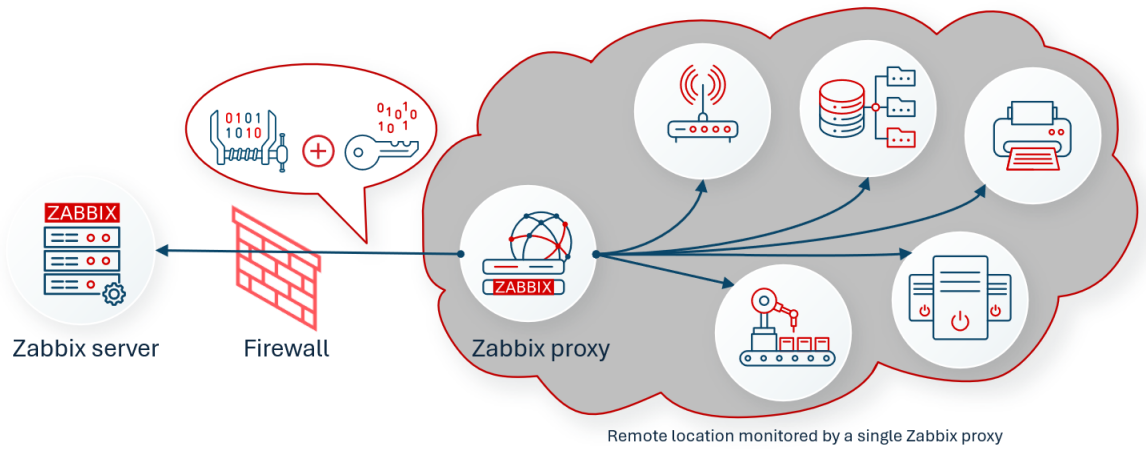
Обзор

Прокси Zabbix можно использовать для создания эффективной и надежной распределенной системы мониторинга.

Прокси развертываются локально для сбора данных о производительности и доступности. Затем данные передаются на централизованный сервер.

Прокси Zabbix можно использовать для:

- Мониторинга удаленных расположений
- Мониторинга расположений с ненадежной связью
- Снижения нагрузки на сервер Zabbix при мониторинге тысяч устройств
- Сбора телеметрических данных для мониторинга производительности приложений (APM)



Прокси требуется только одно TCP-соединение с сервером Zabbix. Это упрощает обход межсетевого экрана, поскольку необходимо настроить только одно правило межсетевого экрана.

> **Attention:**
> Прокси Zabbix должен использовать отдельную базу данных. Указание базы данных сервера Zabbix приведет к повреждению конфигурации.

Все данные, собранные прокси, сохраняются локально перед передачей на сервер. Таким образом, данные не теряются из-за временных проблем со связью с сервером. Параметры `ProxyLocalBuffer` и `ProxyOfflineBuffer` в файле конфигурации прокси определяют, как долго данные хранятся локально.

Attention:
Может возникнуть ситуация, когда прокси, получающий последние изменения конфигурации непосредственно из базы данных сервера Zabbix, имеет более актуальную конфигурацию, чем сервер Zabbix, конфигурация которого может обновляться не так быстро из-за значения параметра CacheUpdateFrequency. В результате прокси может начать собирать данные и отправлять их на сервер Zabbix, который игнорирует эти данные.

Прокси Zabbix является сборщиком данных. Он не вычисляет триггеры, не обрабатывает события и не отправляет оповещения. Обзор функций, поддерживаемых прокси, представлен в следующей таблице:

Функция		Поддерживается прокси
Элементы данных		
	Проверки агентом Zabbix	**Да**
	Проверки агентом Zabbix (активные)	**Да** [1]
	Простые проверки	**Да**
	Элементы данных trapper	**Да**
	Проверки SNMP	**Да**
	SNMP-трапы	**Да**
	Проверки IPMI	**Да**
	Проверки JMX	**Да**
	Мониторинг файлов журналов	**Да**
	Внутренние проверки	**Да**
	Проверки SSH	**Да**
	Проверки Telnet	**Да**
	Внешние проверки	**Да**
	Зависимые элементы данных	**Да**
	Элементы данных-скрипты	**Да**
	Элементы данных браузера	**Да**
	Элементы данных запросов телеметрии	**Да**
Встроенный веб-мониторинг		**Да**
Предварительная обработка значений элементов данных		**Да**
Обнаружение сети		**Да**
Автоматическая регистрация активных агентов		**Да**
Низкоуровневое обнаружение		**Да** [2]
Удаленные команды		**Да**
Сбор телеметрических данных (APM)		**Да** [3]
Вычисление триггеров		*Нет*
Обработка событий		*Нет*
Корреляция событий		*Нет*
Отправка оповещений		*Нет*

Note:
[1] Чтобы агент запрашивал активные проверки у прокси, а не у сервера, прокси должен быть указан в параметре **ServerActive** в файле конфигурации агента.
 [2] При низкоуровневом обнаружении прокси Zabbix только собирает и предварительно обрабатывает данные, а затем отправляет их на сервер Zabbix для дальнейшей обработки.
 [3] Телеметрические данные собираются только в том случае, если на прокси включен APM (см. вкладку *APM* в разделе конфигурации прокси); сервер Zabbix не может получать телеметрические данные.

Возможности прокси

При выборе использования или неиспользования прокси необходимо учитывать несколько факторов.

	Proxy
Легковесный	**Да**
GUI	Нет
Работает независимо	**Да**
Простое обслуживание	**Да**
Автоматическое создание БД	**Да**[1]
Локальное администрирование	Нет
Готов для встроенного оборудования	**Да**
Однонаправленные TCP-соединения	**Да**

	Proxy
Централизованная конфигурация	**Да**
Генерирует уведомления	Нет

[1] Функция автоматического создания БД работает только с SQLite. Для других поддерживаемых баз данных требуется ручная настройка.

Attention:
Прокси Zabbix не учитывает периоды обслуживания; см. Расчет очередей во время обслуживания для подробностей.

Защита от перегрузки

Если сервер Zabbix был недоступен некоторое время, а прокси собрали много данных, и затем сервер запускается, он может оказаться перегружен (использование history cache некоторое время остается на уровне 95-100%). Такая перегрузка может привести к снижению производительности, когда проверки обрабатываются медленнее, чем должны. Для предотвращения этой ситуации была реализована защита, чтобы избежать проблем, возникающих из-за перегрузки history cache.

Когда history cache сервера Zabbix заполнен, запись в history cache начинает ограничиваться, что приостанавливает процессы сбора данных сервером. Наиболее распространенный случай перегрузки history cache - это период после простоя сервера, когда прокси загружают собранные данные. Чтобы избежать этого, было добавлено ограничение для прокси (в настоящее время его нельзя отключить).

Когда использование history cache достигает 80%, сервер Zabbix переходит в режим ограничения. В режиме ограничения сервер принимает данные от прокси только когда использование history cache ниже 60%, поочередно обслуживая принятые прокси. Как только использование history cache опускается ниже 20%, сервер возвращается в нормальный режим.

Кроме того, в нормальном режиме сервер Zabbix ограничивает отдельные прокси, отправляющие очень большие пакеты (10 000+ записей), если использование history cache превышает 60%. Это решение принимается в момент, когда сервер оценивает загрузку от прокси, и поэтому не всегда сразу отражается на графиках использования history cache (внутренний элемент данных `zabbix[wcache,history,pused]` и интервал его обновления могут не зафиксировать кратковременные пики).

Этот режим ограничения будет продолжаться до тех пор, пока либо использование cache снова не достигнет 80%, либо не снизится до 20%, либо список ограничения не станет пустым. В первом случае сервер снова перестанет принимать данные от прокси. В двух других случаях сервер начнет работать нормально, принимая данные от всех прокси.

Приведенную выше информацию можно проиллюстрировать следующей таблицей:

Использование записи в cache истории	Режим сервера Zabbix	Действие сервера Zabbix
Достигает 80%	Ожидание	Прекращает принимать данные от прокси, но сохраняет *список ограничения* (приоритетный список прокси, к которым нужно обратиться позже).
Достигает 60%	Нормальный, но готов к ограничению	Может отклонять очень большие загрузки от прокси (более 10 тыс. записей) при принятии решения о приеме данных; продолжает принимать другие данные от прокси.
Снижается до 20%	Нормальный	Удаляет список ограничения и начинает принимать данные от прокси в обычном режиме.

Для сопоставления этого поведения сервера Zabbix с метрикой можно использовать внутренний элемент данных zabbix[wcache,history,pused].

Конфигурация

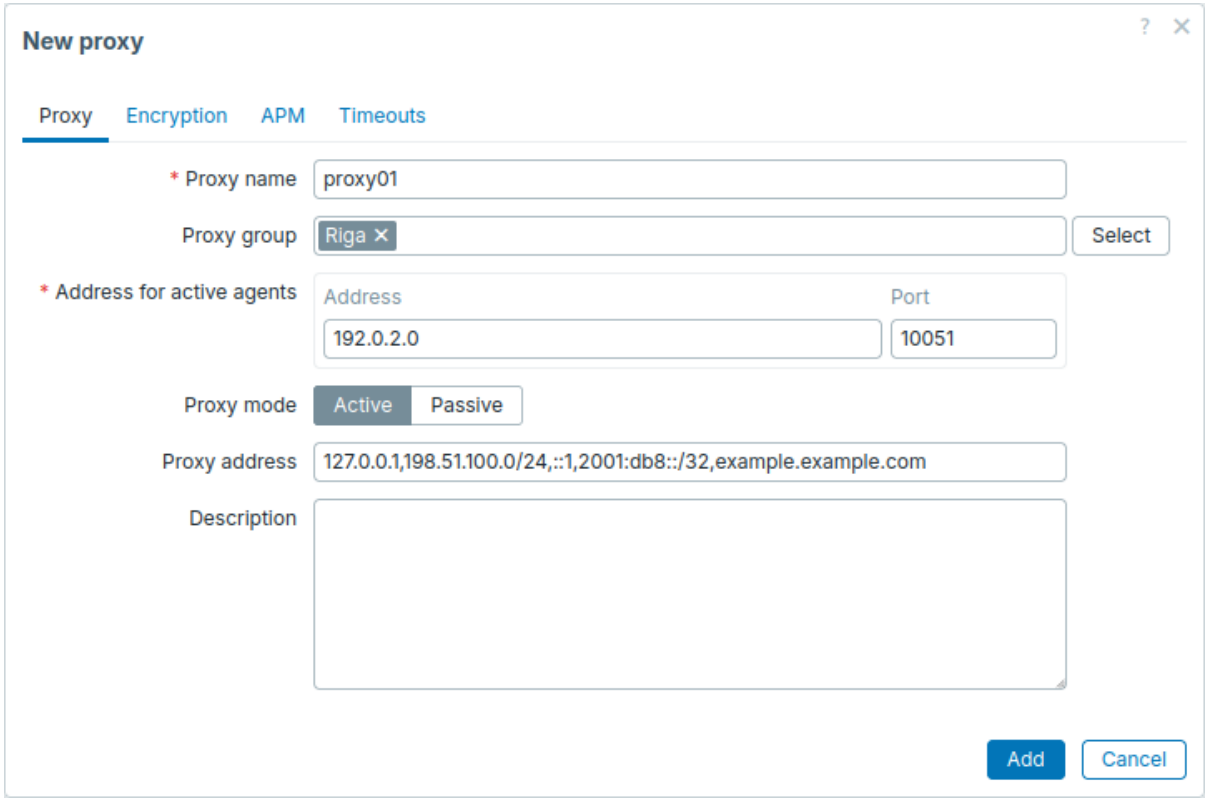
После того как вы установили и настроили прокси, пришло время настроить его во веб-интерфейсе Zabbix.

Добавление прокси

Note:
Вместо установки прокси в собственной инфраструктуре можно развернуть его в Zabbix Cloud. Для этого нажмите *Create proxy in Cloud* в разделе *Administration* > *Proxies*; Zabbix Cloud откроется в новой вкладке браузера. После создания узла прокси настройте прокси в веб-интерфейсе Zabbix, как описано ниже. Дополнительную информацию см. в разделе Настройка прокси Zabbix Cloud.

Чтобы настроить прокси в веб-интерфейсе Zabbix:

1. Перейдите в раздел *Administration* > *Proxies*.
2. Нажмите *Create proxy*.
3. Укажите параметры прокси в форме.



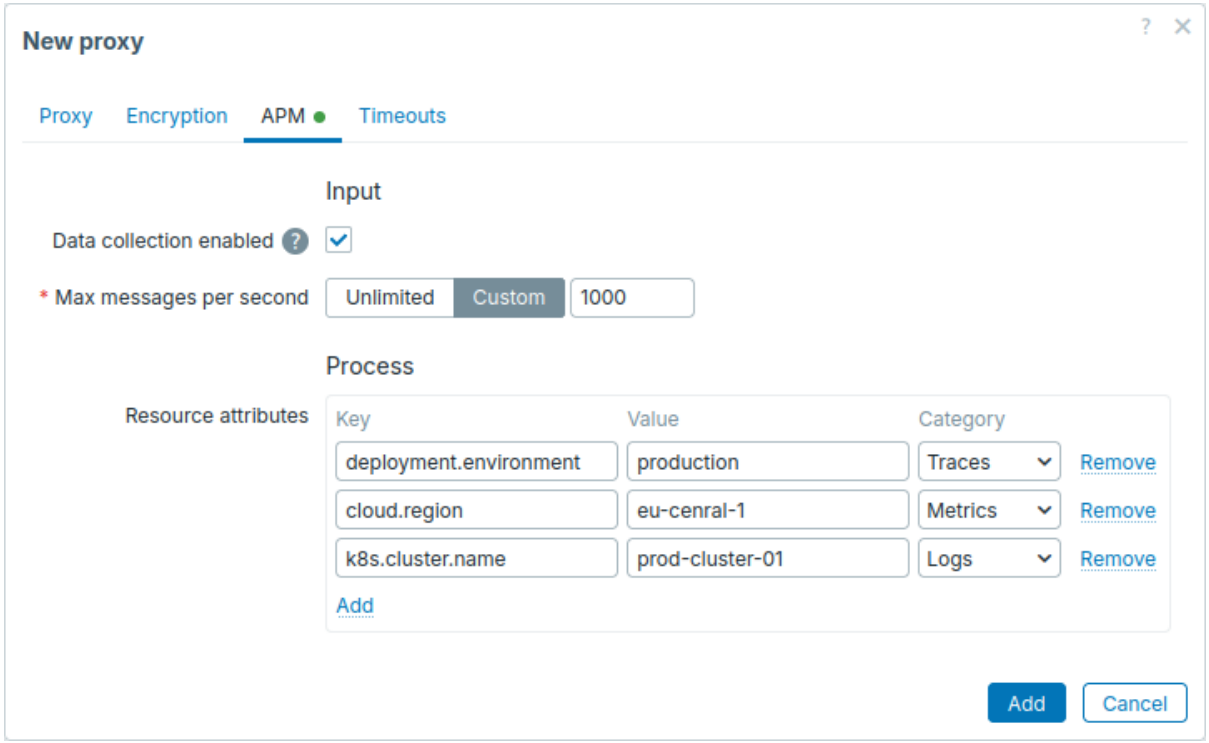
Parameter	Description
Proxy name	Укажите имя прокси. Оно должно совпадать с именем в параметре *Hostname* файла конфигурации прокси.
Proxy group	Выберите одну группу прокси для балансировки нагрузки/высокой доступности прокси.
Address for active agents	Укажите адрес, к которому должны подключаться отслеживаемые активные агенты или отправители. Поддерживается **только** агентами Zabbix версии 7.0 и выше. Этот адрес используется для подключения как к активным, так и к пассивным прокси. Это поле доступно только в том случае, если в поле *Proxy group* выбрана группа прокси.
Address	IP-адрес/DNS-имя для подключения.
Port	Номер TCP-порта (по умолчанию 10051) для подключения. Поддерживаются пользовательские макросы.
Proxy mode	Выберите режим прокси. **Active** - прокси подключается к серверу Zabbix и запрашивает данные конфигурации **Passive** - сервер Zabbix подключается к прокси *Обратите внимание*, что при использовании активного прокси без шифрования соединений (конфиденциальных данных) данные конфигурации прокси могут стать доступными сторонам, имеющим доступ к порту траппера сервера Zabbix. Это возможно, поскольку любой пользователь может выдать себя за активный прокси и запросить данные конфигурации, если аутентификация не выполняется или адреса прокси не ограничены в поле *Proxy address*.

Parameter	Description
Proxy address	Если указано, запросы от активного прокси принимаются только из этого списка IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен активного прокси Zabbix. Это поле доступно только в том случае, если в поле *Proxy mode* выбран активный прокси. Макросы не поддерживаются.
Interface	Укажите сведения об интерфейсе пассивного прокси. Это поле доступно только в том случае, если в поле *Proxy mode* выбран пассивный прокси.
Address	IP-адрес/DNS-имя пассивного прокси.
Port	Номер TCP-порта пассивного прокси (по умолчанию 10051). Поддерживаются пользовательские макросы.
Description	Укажите описание прокси.

Вкладка **Encryption** позволяет потребовать зашифрованные соединения с прокси.

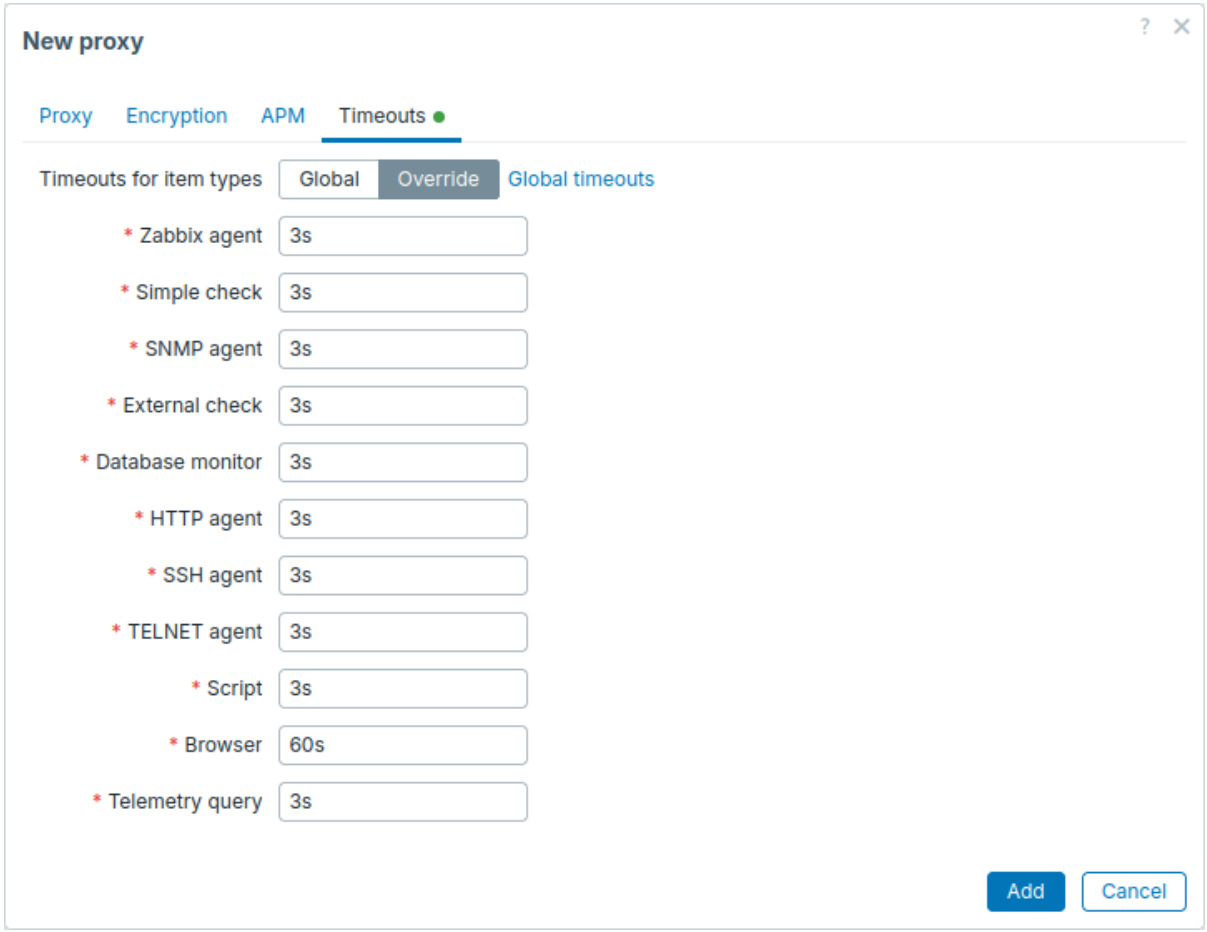
Parameter	Description
Connections to proxy	Способ подключения сервера к пассивному прокси: без шифрования (по умолчанию), с использованием PSK (предварительно установленного ключа) или сертификата.
Connections from proxy	Выберите типы соединений, разрешенных от активного прокси. Одновременно можно выбрать несколько типов соединений (это удобно для тестирования и перехода на другой тип соединения). По умолчанию выбрано «No encryption».
Issuer	Разрешенный издатель сертификата. Сначала сертификат проверяется с помощью CA (центра сертификации). Если сертификат действителен и подписан CA, поле *Issuer* можно использовать для дополнительного ограничения разрешенных CA. Это поле необязательно и предназначено для случаев, когда в установке Zabbix используются сертификаты от нескольких CA.
Subject	Разрешенная тема сертификата. Сначала сертификат проверяется с помощью CA. Если сертификат действителен и подписан CA, поле *Subject* можно использовать, чтобы разрешить только одно значение строки *Subject*. Если это поле пусто, принимается любой действительный сертификат, подписанный настроенным CA.
PSK identity	Строка идентификатора предварительно установленного ключа. Не указывайте конфиденциальную информацию в идентификаторе PSK: он передается по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK следует использовать.
PSK	Предварительно установленный ключ (шестнадцатеричная строка). Максимальная длина: 512 шестнадцатеричных цифр (PSK размером 256 байт), если Zabbix использует библиотеку GnuTLS или OpenSSL, и 64 шестнадцатеричные цифры (PSK размером 32 байта), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Вкладка **APM** позволяет настроить мониторинг производительности приложений (APM) на прокси. Прокси может собирать телеметрические данные (трассировки, метрики и журналы), отправляемые приложениями или коллекторами OpenTelemetry по протоколу OpenTelemetry (OTLP/gRPC), и записывать эти данные в источник телеметрических данных.



Parameter	Description
Data collection enabled	Установите флажок, чтобы включить сбор телеметрических данных на прокси. Перед включением сбора телеметрических данных установите параметр `StartAPMCollector` в значение 1 в файле конфигурации прокси (требуется прокси Zabbix с поддержкой APM).
Max messages per second	Укажите максимальное количество сообщений в секунду, принимаемых прокси: **Unlimited** - принимать все входящие сообщения. **Custom** - принимать до указанного количества сообщений в секунду. Сообщение представляет собой запрос OTLP, отправленный приложением, коллектором OpenTelemetry или другим источником телеметрии. Его размер зависит от отправителя; например, коллектор OpenTelemetry может объединить тысячи спанов в одно сообщение, тогда как простое приложение может отправлять сообщения меньшего размера (например, по одному на трассировку). Если ограничение превышено, прокси отвечает кодом состояния gRPC `UNAVAILABLE` (временная ошибка), поэтому отправитель может повторить отправку сообщения позднее в зависимости от своей конфигурации.
Add resource attributes	Укажите атрибуты ресурса, которые следует добавить во входящие телеметрические данные выбранного типа (например, чтобы пометить все данные, полученные этим прокси, информацией о среде или регионе). Затем эти атрибуты можно использовать для фильтрации телеметрических данных, например в условиях элементов данных запроса телеметрии. Если входящие данные уже содержат атрибут ресурса с таким же именем, сохраняется входящий атрибут, а не указанный здесь.
Name	Укажите имя атрибута (например, `deployment.environment`, `cloud.region`). Имя должно быть уникальным для выбранного значения *Type*.
Value	Укажите значение атрибута (например, `production`, `eu-central-1`). Может быть пустым.
Type	Выберите тип входящих данных, к которым применяется атрибут: *Traces*, *Metrics* или *Logs*.

Вкладка **Timeouts** позволяет переопределить глобальные тайм-ауты для поддерживающих их типов элементов данных.



Parameter	Description
Timeouts for item types	Укажите тайм-аут элемента данных (в зависимости от его типа): **Global** - использовать глобальный тайм-аут (отображается в неактивном поле *Timeout* для каждого типа элемента данных); **Override** - задать пользовательский тайм-аут (в поле *Timeout* для каждого типа элемента данных). Допустимый диапазон: 1 - 600 с (по умолчанию: значение наследуется из глобальных тайм-аутов). Поддерживаются суффиксы времени, например 30s, 1m, и пользовательские макросы. Нажатие ссылки *Global timeouts* позволяет настроить глобальные тайм-ауты. Обратите внимание, что ссылка *Global timeouts* видна только пользователям типа *Super admin*, имеющим разрешения на раздел веб-интерфейса *Administration* > *General*. Обратите внимание, что тайм-ауты на уровне прокси переопределяют глобальные, но сами переопределяются индивидуальными тайм-аутами элементов данных, если они настроены.

Note:
Если основной номер версии прокси не совпадает с основным номером версии сервера, рядом с полем *Timeouts for item types* отображается значок с сообщением при наведении: «Timeouts disabled because the proxy and server versions do not match». В таких случаях прокси использует параметр `Timeout` из файла конфигурации прокси.

Форма редактирования существующего прокси содержит следующие дополнительные кнопки:

- *Refresh configuration* - обновить конфигурацию прокси;
- *Clone* - создать новый прокси на основе свойств существующего прокси;
- *Delete* - удалить прокси.

Конфигурация узла сети

Вы можете указать, что отдельный узел сети должен отслеживаться прокси или группой прокси, в форме конфигурации узла сети, используя поле *Monitored by*.



Массовое обновление узлов сети — это еще один способ указать, что узлы сети должны отслеживаться прокси или группой прокси.

Балансировка нагрузки и высокая доступность прокси

Обзор

Прокси Zabbix можно объединять в группы прокси для балансировки нагрузки и обеспечения высокой доступности прокси.

Балансировка нагрузки и обеспечение высокой доступности прокси — это автоматическое перераспределение узлов сети между прокси внутри группы прокси:

- Если прокси переходит в автономный режим, его узлы сети перемещаются на другие прокси, что позволяет поддерживать высокую доступность прокси.
- Если на прокси настроено значительно больше или меньше узлов сети, чем на других прокси, его узлы сети перемещаются на другие прокси для балансировки нагрузки.

Перераспределение узлов сети работает только между прокси в группе, которые соответствуют следующим условиям:

- На прокси установлена версия Zabbix 7.0 или более поздняя.
- Версия прокси совпадает с версией сервера Zabbix. При использовании агента Zabbix (пассивного) версия прокси должна совпадать с версией агента. Для активных агентов требуется только версия Zabbix 7.0 или более поздняя.
- Группа прокси имеет состояние «В сети».
- Для узлов сети настроен мониторинг через группу прокси, а не через отдельные прокси.

Note:
Состояние группы прокси можно отслеживать с помощью внутренних проверок на любом узле сети, назначенном группе прокси. Однако для отслеживания состояния отдельного прокси в группе назначьте узел сети этому прокси; в противном случае результаты могут быть непоследовательными.

Перераспределение узлов сети

Балансировка нагрузки прокси и высокая доступность управляются сервером Zabbix через менеджер группы прокси, который непрерывно отслеживает состояние всех прокси в каждой группе прокси и распределение их узлов сети.

Высокая доступность прокси внутри группы обеспечивается за счет отказоустойчивости прокси: когда прокси переходит в автономный режим, его узлы сети немедленно перераспределяются между другими прокси. Также выполняется балансировка нагрузки прокси, поскольку узлы сети переназначаются на прокси с наименьшим количеством назначенных узлов сети.

Кроме того, балансировка нагрузки прокси запускается, когда количество узлов сети на прокси отличается от среднего по группе как минимум на 10 узлов сети и в 2 раза (избыток узлов сети или дефицит узлов сети). Если дисбаланс сохраняется после периода ожидания (10 x задержка отказоустойчивости), группа прокси ставится в очередь на перераспределение узлов сети.

Менеджер группы прокси перераспределяет узлы сети по следующей логике:

1. Вычислить среднее количество узлов сети на один прокси.
2. Для прокси с избытком узлов сети — переместить лишние узлы сети в пул не назначенных прокси.
3. Для прокси с дефицитом узлов сети — вычислить, сколько узлов сети нужно для достижения баланса.
4. Удалить необходимое количество узлов сети с прокси, у которых их больше всего.
5. Переместить не назначенные узлы сети на прокси с наименьшим количеством узлов сети.

Примеры перераспределения узлов сети:

Узлов сети на прокси	Среднее по группе	Переназначение узлов сети
100	50	Да
60	50	Нет
40	50	Нет
25	50	Да
15	5	Да

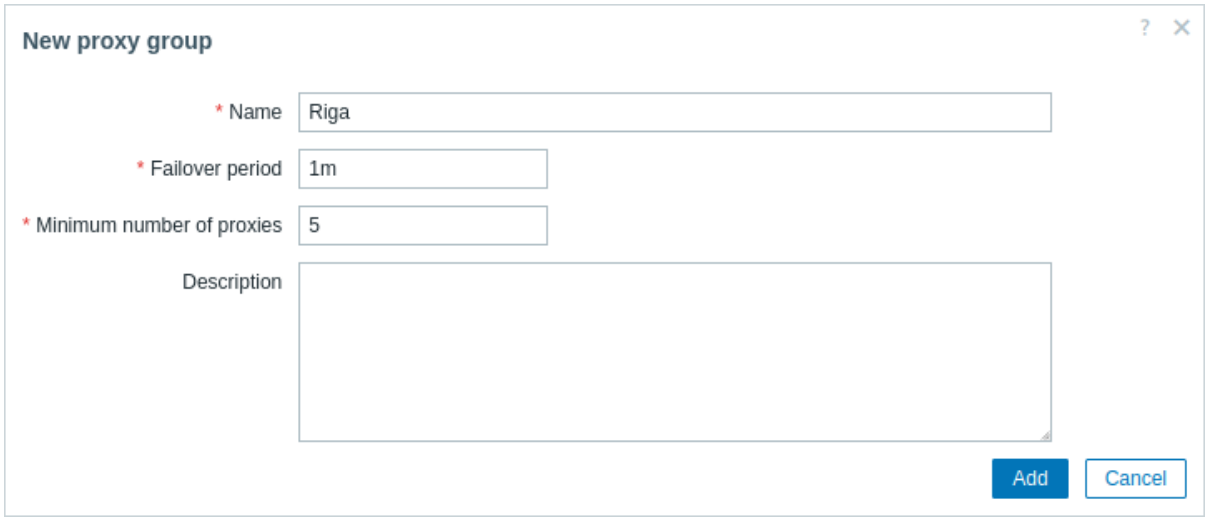
Узлов сети на прокси	Среднее по группе	Переназначение узлов сети
10	5	Нет

Attention:
Наличие менее 10 узлов сети, контролируемых группой прокси, может привести к неравномерному распределению узлов сети между прокси в группе.

Настройка группы прокси

Чтобы настроить группу прокси в веб-интерфейсе Zabbix:

1. Перейдите в раздел *Администрирование* > *Группы прокси*.
2. Нажмите *Создать группу прокси*.
3. Укажите параметры группы прокси в форме.



Параметр	Описание
Имя	Имя группы прокси.
Период переключения	Период в секундах, в течение которого прокси в группе прокси должен обмениваться данными с сервером Zabbix, чтобы считаться доступным (по умолчанию: 1m; диапазон: 10s-15m). Если прокси не обменивается данными в течение этого периода, его состояние изменяется на *Недоступен*, а узлы сети, назначенные ему, немедленно перераспределяются между другими прокси. Балансировка нагрузки прокси начинается после истечения периода, умноженного на 10. Поддерживаются суффиксы времени (например, 30s, 1m) и пользовательские макросы.
Минимальное количество прокси	Минимальное количество доступных прокси, необходимое для сохранения доступности группы прокси (по умолчанию: 1; диапазон: 1-1000). Поддерживаются пользовательские макросы. Это значение должно быть меньше общего количества прокси в группе. Например, если в группе 10 прокси, установка минимального значения 10 приведет к тому, что группа станет недоступной при сбое любого прокси. Обратите внимание, что доступные прокси в недоступной группе продолжают работать в обычном режиме, однако балансировка нагрузки и высокая доступность работать не будут.
Описание	Описание группы прокси.
Прокси	При редактировании группы, содержащей прокси, отображается список из не более чем пяти прокси (в виде ссылок или обычного текста в зависимости от прав пользователя на прокси).

Настройка балансировки нагрузки прокси

Чтобы использовать балансировку нагрузки прокси, необходимо настроить группу прокси в веб-интерфейсе Zabbix (см. выше) и убедиться, что узлы сети контролируются группой прокси, а не отдельными прокси (для перемещения узлов сети от прокси в группу прокси можно использовать массовое обновление узлов сети).

При использовании агента Zabbix также настройте его следующим образом:

- Для пассивных проверок укажите все прокси группы прокси в параметре Server.
- Для активных проверок рекомендуется указать все прокси группы прокси или сервер Zabbix в параметре ServerActive. Обратите внимание, что с группами прокси в активном режиме будет работать только агент Zabbix версии 7.0 или более поздней.

Если параметр `ServerActive` содержит только один прокси из группы прокси (или сервер Zabbix), агент всё равно сможет подключиться к правильному прокси. При запуске службы агента и подключении к указанному прокси агент получит и кэширует полный список IP-адресов прокси и их текущую нагрузку в группе. Затем активные проверки будут перенаправляться к правильному доступному прокси для узла сети на основе текущего назначения узлов сети прокси внутри группы прокси.

Warning:
Если в параметре `ServerActive` агента Zabbix указан только один прокси, это может привести к потере данных мониторинга, если агент запускается или перезапускается в то время, когда указанный прокси недоступен.

Note:
При использовании Zabbix sender запросы данных также перенаправляются к правильному доступному прокси для узла сети на основе текущего назначения узлов сети прокси внутри группы прокси. Однако при отправке значений нескольких узлов сети из входного файла используйте опцию `-g`, чтобы предотвратить отправку данных неправильному прокси.

Агент Zabbix также должен иметь возможность подключаться ко всем прокси в группе прокси через межсетевой экран. В противном случае активные проверки могут зависать или завершаться ошибкой во время перенаправления или переключения на резервный прокси. Например:

- Во время активных проверок прокси может перенаправить агента к другому прокси. Если этот прокси заблокирован межсетевым экраном, обмен данными зависнет в ожидании ответа.
- В стабильных конфигурациях с высокой доступностью без недавнего перераспределения агенты могут никогда не обращаться к резервным прокси. Если правила межсетевого экрана изменились и не были протестированы, переключение на резервный прокси может завершиться ошибкой.

Тестирование балансировки нагрузки прокси

Чтобы протестировать балансировку нагрузки прокси:

1. Настройте группу прокси.
2. Убедитесь, что группа прокси имеет состояние «в сети».
3. Убедитесь, что узлы сети контролируются группой прокси, а не отдельными прокси (для перемещения узлов сети от прокси в группу прокси можно использовать массовое обновление).
4. Подождите несколько секунд, пока обновится конфигурация и узлы сети распределятся между прокси в группе прокси.

Проследите за изменениями, обновив список узлов сети в разделе *Администрирование* > *Прокси*.

Важные примечания

- SNMP-трапы не поддерживаются прокси в группе прокси.
- Проверки, зависящие от внешней конфигурации (например, скрипты для внешних проверок или конфигурация ODBC для проверок базы данных), должны иметь одинаковую конфигурацию на всех прокси в группе прокси.
- Для проверок базы данных требуются расширенные разрешения на объект базы данных или сервер.
- Узлы VMware, мониторинг которых выполняется группой прокси, будут случайным образом распределены между прокси в группе. Это приводит к тому, что каждый прокси кэширует все данные VMware, создавая дополнительную нагрузку на vCenter.
- Для узлов сети, созданных на основе данных автоматической регистрации от прокси в группе прокси, в качестве объекта, осуществляющего мониторинг, устанавливается эта группа прокси. Однако для узлов сети, созданных на основе данных сетевого обнаружения от прокси в группе прокси, в качестве объекта, осуществляющего мониторинг, устанавливается этот прокси.

Синхронизация конфигурации мониторинга

Обзор

На этой странице приведены сведения об обновлении конфигурации мониторинга для прокси, то есть о том, как изменения, внесенные в конфигурацию мониторинга на сервере, синхронизируются с прокси.

Инкрементальное обновление

Обновление конфигурации прокси является инкрементальным.
Во время синхронизации конфигурации обновляются только измененные сущности (поэтому, если ни одна сущность не была изменена, ничего отправлено не будет).
Такой подход позволяет экономить ресурсы и устанавливать меньший интервал (почти мгновенный) для обновления конфигурации прокси.

Изменения конфигурации прокси отслеживаются с помощью номеров ревизий. В данные конфигурации, отправляемые прокси, включаются только сущности с номерами ревизий, превышающими номер ревизии конфигурации прокси.

Сущности, участвующие в синхронизации конфигурации:

Сущность	Подробности
autoregistration tls data	Все данные TLS для автоматической регистрации.
expressions	Все выражения (регулярные выражения, таблицы выражений).
global configuration	Глобальная конфигурация, определенная в таблице 'config'
host	Все свойства, интерфейсы, данные инвентаризации, элементы данных, предобработка элементов данных, параметры элементов данных и веб-сценарии узла сети.
host macros	Все макросы, определенные на узле сети, и все идентификаторы шаблонов, связанные с ним.
proxy discovery rule	Правила обнаружения и проверки, назначенные прокси.

Это означает следующее:

- Если элемент данных изменен на **узле сети**, вся конфигурация этого узла сети будет синхронизирована.
- Если изменено **регулярное выражение**, все регулярные выражения будут синхронизированы.

Исключением являются макросы узла сети: они также отправляются, если на узле сети было изменено что-либо еще.

Команда `-R config_cache_reload` на прокси также инициирует инкрементальное обновление.

Обратите внимание, что полная синхронизация конфигурации выполняется при запуске или перезапуске прокси, переключении HA, изменении токена сеанса или сбое обновления конфигурации на прокси, например если соединение было разорвано во время получения данных конфигурации.

Параметры конфигурации

Параметр **ProxyConfigFrequency** определяет, как часто конфигурация прокси синхронизируется с сервером (по умолчанию каждые 10 секунд).

Обратите внимание, что ProxyConfigFrequency является:

- параметром сервера для пассивных прокси
- параметром прокси для активных прокси

Для активных прокси ProxyConfigFrequency — это новый параметр, начиная с Zabbix 6.4, и его следует использовать вместо устаревшего ConfigFrequency.

> **Attention:**
> Если используются и ProxyConfigFrequency, и ConfigFrequency, прокси запишет ошибку в журнал и завершит работу.

Расчет очередей во время обслуживания

> **Attention:**
> Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Расчет очередей во время обслуживания.

Протокол обмена данными между сервером и прокси

Сервер - обмен данными с прокси основан на формате JSON.

Сообщения запроса и ответа должны начинаться с заголовка и длины данных.

Пассивный прокси

Запрос конфигурации

Сначала сервер отправляет пустой запрос `proxy config`. Этот запрос отправляется каждые `ProxyConfigFrequency` секунд (параметр конфигурации сервера).

Прокси отвечает, указывая текущую версию прокси, токен сеанса и редакцию конфигурации. Сервер отвечает данными конфигурации, которые необходимо обновить.

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	Версия прокси (<major>.<minor>.<build>).
session	*string*	Токен сеанса конфигурации прокси.
config_revision	*number*	Редакция конфигурации прокси.
server→proxy:		
full_sync	*number*	1 - если отправляются полные данные конфигурации; в противном случае отсутствует (необязательный параметр).
data	*array*	Объект с данными таблиц. Отсутствует, если конфигурация не изменялась (необязательный параметр).
<table>	*object*	Один или несколько объектов с данными <table> (необязательный параметр, зависит от изменений).
fields	*array*	Массив имен полей.
-	*string*	Имя поля.
data	*array*	Массив строк.
-	*array*	Массив столбцов.
-	*string*,*number*	Значение столбца с типом, зависящим от типа столбца в схеме базы данных.
macro.secrets	*object*	Информация о секретных макросах; отсутствует, если в макросах хранилища нет изменений (необязательный параметр).
config_revision	*number*	Редакция кэша конфигурации - отправляется вместе с данными конфигурации (необязательный параметр).
del_hostids	*array*	Массив удаленных hostids (необязательный параметр).
-	*number*	Идентификатор узла сети.
del_macro_hostids	*array*	Массив hostids, у которых удалены все макросы (необязательный параметр).
-	*number*	Идентификатор узла сети.
proxy→server:		
response	*string*	Информация об успешном выполнении запроса ('success' или 'failed').
version	*string*	Версия прокси (<major>.<minor>.<build>).

Пример:

сервер→прокси:

```
{
  "request":"proxy config"
}
```

прокси→сервер:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

сервер→прокси:

```
{
    "full_sync": 1,
    "data": {
        "hosts": {
```

```
            "fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
                [44162, 0, 0]
            ]
        },
        "item_preproc": {
            "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
            "data": []
        },
        "item_parameter": {
            "fields": ["item_parameterid", "itemid", "name", "value"],
            "data": []
        },
        "globalmacro": {
            "fields": ["globalmacroid", "macro", "value", "type"],
            "data": [
                [2, "{$SNMP_COMMUNITY}", "public", 0]
            ]
        },
        "hosts_templates": {
            "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
            "data": []
        },
        "hostmacro": {
            "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
            "data": [
                [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
            ]
        },
        "drules": {
            "fields": ["druleid", "name", "iprange", "delay"],
            "data": [
```

```
            [2, "Local network", "127.0.0.1", "10s"]
        ]
    },
    "dchecks": {
        "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
        "data": [
            [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
        ]
    },
    "regexps": {
        "fields": ["regexpid", "name"],
        "data": [
            [1, "File systems for discovery"],
            [2, "Network interfaces for discovery"],
            [3, "Storage devices for SNMP discovery"],
            [4, "Windows service names for discovery"],
            [5, "Windows service startup states for discovery"]
        ]
    },
    "expressions": {
        "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
        "data": [
            [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
            [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
            [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
            [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
            [7, 2, "^Software Loopback Interface", 4, ",", 1],
            [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
            [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
            [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
            [11, 2, "^[Ss]ystem$", 4, ",", 1],
            [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
        ]
    },
    "settings": {
        "fields": ["name", "type", "value_str", "value_int"],
        "data": [
            ["autoreg_tls_accept", 2, "", 1],
            ["hk_history_global", 2, "", 0],
            ["snmptrap_logging", 2, "", 1],
            ["proxy_secrets_provider", 2, "", 0],
            ["hk_history", 1, "31d", 0],
            ["timeout_db_monitor", 1, "3s", 0],
            ["timeout_external_check", 1, "3s", 0],
            ["timeout_http_agent", 1, "3s", 0],
            ["timeout_simple_check", 1, "3s", 0],
            ["timeout_snmp_agent", 1, "3s", 0],
            ["timeout_ssh_agent", 1, "3s", 0],
            ["timeout_telnet_agent", 1, "3s", 0],
            ["timeout_zabbix_agent", 1, "3s", 0],
            ["timeout_browser", 1, "30s", 0]
        ]
    },
    "httptest": {
        "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
        "data": []
    },
    "httptestitem": {
        "fields": ["httptestitemid", "httptestid", "itemid", "type"],
        "data": []
    },
    "httptest_field": {
```

```
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```


прокси→сервер:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Запрос данных

Запрос `proxy data` используется для получения данных о доступности интерфейсов узлов сети, исторических данных, данных обнаружения и данных авто-регистрации от прокси. Этот запрос отправляется каждые `ProxyDataFrequency` (параметр конфигурации сервера) секунд.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Токен сеанса данных.
interface availability	*array*	*(optional)* Массив объектов данных о доступности интерфейсов.
interfaceid	*number*	Идентификатор интерфейса.
available	*number*	Доступность интерфейса: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - неизвестно **1**, *INTERFACE_AVAILABLE_TRUE* - доступен **2**, *INTERFACE_AVAILABLE_FALSE* - недоступен
error	*string*	Сообщение об ошибке интерфейса или пустая строка.
history data	*array*	*(optional)* Массив объектов исторических данных.
itemid	*number*	Идентификатор элемента данных.
clock	*number*	Метка времени значения элемента данных (секунды).
ns	*number*	Метка времени значения элемента данных (наносекунды).

name	value type	description
value	*string*	*(optional)* Значение элемента данных.
id	*number*	Идентификатор значения (возрастающий счетчик, уникальный в пределах одного сеанса данных).
timestamp	*number*	*(optional)* Метка времени для элементов данных типа log.
source	*string*	*(optional)* Значение источника элемента данных eventlog.
severity	*number*	*(optional)* Значение уровня серьезности элемента данных eventlog.
eventid	*number*	*(optional)* Значение eventid элемента данных eventlog.
state	*string*	*(optional)* Состояние элемента данных: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Последний размер журнала для элементов данных типа log.
mtime	*number*	*(optional)* Время изменения для элементов данных типа log.
discovery data	*array*	*(optional)* Массив объектов данных обнаружения.
clock	*number*	Метка времени данных обнаружения.
druleid	*number*	Идентификатор правила обнаружения.
dcheckid	*number*	Идентификатор проверки обнаружения или null для данных правила обнаружения.
type	*number*	Тип проверки обнаружения: **-1** данные правила обнаружения **0**, *SVC_SSH* - проверка службы SSH **1**, *SVC_LDAP* - проверка службы LDAP **2**, *SVC_SMTP* - проверка службы SMTP **3**, *SVC_FTP* - проверка службы FTP **4**, *SVC_HTTP* - проверка службы HTTP **5**, *SVC_POP* - проверка службы POP **6**, *SVC_NNTP* - проверка службы NNTP **7**, *SVC_IMAP* - проверка службы IMAP **8**, *SVC_TCP* - проверка доступности TCP-порта **9**, *SVC_AGENT* - агент Zabbix **10**, *SVC_SNMPv1* - агент SNMPv1 **11**, *SVC_SNMPv2* - агент SNMPv2 **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - агент SNMPv3 **14**, *SVC_HTTPS* - проверка службы HTTPS **15**, *SVC_TELNET* - проверка доступности Telnet
ip	*string*	IP-адрес узла сети.
dns	*string*	DNS-имя узла сети.
port	*number*	*(optional)* Номер порта службы.
key_	*string*	*(optional)* Ключ элемента данных для проверки обнаружения типа **9** *SVC_AGENT*
value	*string*	*(optional)* Значение, полученное от службы; для большинства служб может быть пустым.
status	*number*	*(optional)* Состояние службы: **0**, *DOBJECT_STATUS_UP* - служба доступна **1**, *DOBJECT_STATUS_DOWN* - служба недоступна
auto registration	*array*	*(optional)* Массив объектов данных авто-регистрации.
clock	*number*	Метка времени данных авто-регистрации.
host	*string*	Имя узла сети.
ip	*string*	*(optional)* IP-адрес узла сети.
dns	*string*	*(optional)* Разрешенное DNS-имя по IP-адресу.
port	*string*	*(optional)* Порт узла сети.
host_metadata	*string*	*(optional)* Метаданные узла сети, отправленные агентом (на основе параметра конфигурации агента HostMetadata или HostMetadataItem).
tasks	*array*	*(optional)* Массив задач.

name	value type	description
type	*number*	Тип задачи: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - результат удаленной команды
status	*number*	Статус выполнения удаленной команды: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - удаленная команда успешно выполнена **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - удаленная команда не выполнена
error	*string*	*(optional)* Сообщение об ошибке.
parent_taskid	*number*	ID родительской задачи.
more	*number*	*(optional)* 1 - требуется отправить еще исторические данные.
clock	*number*	*(optional)* Метка времени передачи данных (секунды).
ns	*number*	*(optional)* Метка времени передачи данных (наносекунды).
version	*string*	Версия прокси (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Информация об успешности запроса ('success' или 'failed').
tasks	*array*	*(optional)* Массив задач.
type	*number*	Тип задачи: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - удаленная команда
clock	*number*	Время создания задачи.
ttl	*number*	Время в секундах, после которого задача истекает.
commandtype	*number*	Тип удаленной команды: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - использовать пользовательский скрипт **1**, *ZBX_SCRIPT_TYPE_IPMI* - использовать IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - использовать SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - использовать Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - использовать глобальный скрипт (в настоящее время функционально эквивалентен пользовательскому скрипту)
command	*string*	Удаленная команда для выполнения.
execute_on	*number*	Цель выполнения для пользовательских скриптов: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - выполнить скрипт на агенте **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - выполнить скрипт на сервере **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - выполнить скрипт на прокси
port	*number*	*(optional)* Порт для команд Telnet и SSH.
authtype	*number*	*(optional)* Тип аутентификации для команд SSH.
username	*string*	*(optional)* Имя пользователя для команд Telnet и SSH.
password	*string*	*(optional)* Пароль для команд Telnet и SSH.
publickey	*string*	*(optional)* Открытый ключ для команд SSH.
privatekey	*string*	*(optional)* Закрытый ключ для команд SSH.
parent_taskid	*number*	ID родительской задачи.
hostid	*number*	ID целевого узла сети.

Пример:

server→proxy:

```
{
  "request": "proxy data"
}
```

proxy→server:

```
{
    "session": "12345678901234567890123456789012"
    "interface availability": [
```

```
        {
            "interfaceid": 1,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 2,
            "available": 2,
            "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
    },
        {
            "interfaceid": 3,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 4,
            "available": 1,
            "error": ""
    }
    ],
    "history data":[
        {
            "itemid":"12345",
            "clock":1478609647,
            "ns":332510044,
            "value":"52956612",
            "id": 1
        },
        {
            "itemid":"12346",
            "clock":1478609647,
            "ns":330690279,
            "state":1,
            "value":"Cannot find information for this network interface in /proc/net/dev.",
            "id": 2
        }
    ],
    "discovery data":[
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":3,
            "type":12,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        },
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":null,
            "type":-1,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        }
    ],
    "auto registration":[
        {
            "clock":1478608371,
            "host":"Logger1",
```

```
            "ip":"10.3.0.1",
            "dns":"localhost",
            "port":"10050"
        },
        {
            "clock":1478608381,
            "host":"Logger2",
            "ip":"10.3.0.2",
            "dns":"localhost",
            "port":"10050"
        }
    ],
    "tasks":[
        {
            "type": 0,
            "status": 0,
            "parent_taskid": 10
        },
        {
            "type": 0,
            "status": 1,
            "error": "No permissions to execute task.",
            "parent_taskid": 20
        }
    ],
    "version":"8.0.0"
}
```


server→proxy:


```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
```

```
      }
  ]
}
```

Запрос задач

Обмен `proxy tasks` обеспечивает передачу и подтверждение получения задач удаленных команд между сервером и пассивным прокси. Когда сервер опрашивает прокси на предмет результатов задач, он отправляет пустой запрос `proxy tasks` с интервалом, заданным параметром `ProxyDataFrequency`. Прокси отвечает своей текущей версией и всеми ожидающими результатами задач (включая временные метки). Затем сервер подтверждает получение, возвращая `response`, и может включить новые задачи для выполнения прокси.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Версия прокси (<major>.<minor>.<build>).
clock	*number*	*(optional)* Временная метка передачи данных (секунды).
ns	*number*	*(optional)* Временная метка передачи данных (наносекунды).
server→proxy:		
response	*string*	Информация об успешности запроса ('success' or 'failed').

Пример:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

Активный прокси

Запрос конфигурации

Запрос `proxy config` отправляется активным прокси для получения данных конфигурации прокси. Этот запрос отправляется каждые `ProxyConfigFrequency` секунд (параметр конфигурации прокси).

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	Имя прокси.
version	*string*	Версия прокси (<major>.<minor>.<build>).
session	*string*	Токен сеанса конфигурации прокси.
config_revision	*number*	Ревизия конфигурации прокси.
hostmap_revision	*number*	Текущая ревизия сопоставления узлов сети с прокси.
server→proxy:		
full_sync	*number*	1 - если отправляются полные данные конфигурации; в противном случае отсутствует (необязательный параметр).
data	*array*	Объект с данными таблиц. Отсутствует, если конфигурация не изменялась (необязательный параметр).

name	value type	description
<table>	*object*	Один или несколько объектов с данными <table> (необязательный параметр, зависит от изменений).
fields	*array*	Массив имен полей.
-	*string*	Имя поля.
data	*array*	Массив строк.
-	*array*	Массив столбцов.
-	*string,number*	Значение столбца с типом, зависящим от типа столбца в схеме базы данных.
macro.secrets	*object*	Информация о секретных макросах; отсутствует, если в макросах хранилища нет изменений (необязательный параметр).
proxy_group	*string*	Имя группы прокси, к которой принадлежит прокси.
config_revision	*number*	Ревизия кэша конфигурации - отправляется вместе с данными конфигурации (необязательный параметр).
del_hostids	*array*	Массив удаленных идентификаторов узлов сети (необязательный параметр).
-	*number*	Идентификатор узла сети.
del_macro_hostids	*array*	Массив идентификаторов узлов сети, у которых удалены все макросы (необязательный параметр).
-	*number*	Идентификатор узла сети.

Пример:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
```

```
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
```

```
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
```

```
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Запрос данных прокси

Запрос `proxy data` отправляется прокси для передачи данных о доступности интерфейсов узлов сети, истории, обнаружении и автоматической регистрации. Этот запрос отправляется каждые `DataSenderFrequency` секунд (параметр конфигурации прокси). Обратите внимание, что активный прокси по-прежнему будет опрашивать сервер Zabbix каждую секунду на наличие задач удаленных команд (с пустым запросом `proxy data`).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Имя прокси.
session	*string*	Токен сеанса данных.
interface availability	*array*	*(optional)* Массив объектов данных о доступности интерфейсов.
interfaceid	*number*	Идентификатор интерфейса.
available	*number*	Доступность интерфейса: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - неизвестно **1**, *INTERFACE_AVAILABLE_TRUE* - доступен **2**, *INTERFACE_AVAILABLE_FALSE* - недоступен
error	*string*	Сообщение об ошибке интерфейса или пустая строка.
history data	*array*	*(optional)* Массив объектов данных истории.
itemid	*number*	Идентификатор элемента данных.
clock	*number*	Метка времени значения элемента данных (секунды).
ns	*number*	Метка времени значения элемента данных (наносекунды).
value	*string*	*(optional)* Значение элемента данных.
id	*number*	Идентификатор значения (возрастающий счетчик, уникальный в рамках одного сеанса данных).
timestamp	*number*	*(optional)* Метка времени элементов данных типа log.
source	*string*	*(optional)* Значение источника элемента eventlog.
severity	*number*	*(optional)* Значение уровня серьезности элемента eventlog.
eventid	*number*	*(optional)* Значение eventid элемента eventlog.
state	*string*	*(optional)* Состояние элемента данных: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Последний размер журнала для элементов данных типа log.
mtime	*number*	*(optional)* Время изменения для элементов данных типа log.
discovery data	*array*	*(optional)* Массив объектов данных обнаружения.
clock	*number*	Метка времени данных обнаружения.
druleid	*number*	Идентификатор правила обнаружения.
dcheckid	*number*	Идентификатор проверки обнаружения или null для данных правила обнаружения.

name	value type	description
type	*number*	Тип проверки обнаружения: **-1** данные правила обнаружения **0**, *SVC_SSH* - проверка службы SSH **1**, *SVC_LDAP* - проверка службы LDAP **2**, *SVC_SMTP* - проверка службы SMTP **3**, *SVC_FTP* - проверка службы FTP **4**, *SVC_HTTP* - проверка службы HTTP **5**, *SVC_POP* - проверка службы POP **6**, *SVC_NNTP* - проверка службы NNTP **7**, *SVC_IMAP* - проверка службы IMAP **8**, *SVC_TCP* - проверка доступности TCP-порта **9**, *SVC_AGENT* - агент Zabbix **10**, *SVC_SNMPv1* - агент SNMPv1 **11**, *SVC_SNMPv2* - агент SNMPv2 **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - агент SNMPv3 **14**, *SVC_HTTPS* - проверка службы HTTPS **15**, *SVC_TELNET* - проверка доступности Telnet
ip	*string*	IP-адрес узла сети.
dns	*string*	DNS-имя узла сети.
port	*number*	*(optional)* Номер порта службы.
key_	*string*	*(optional)* Ключ элемента данных для проверки обнаружения типа **9** *SVC_AGENT*
value	*string*	*(optional)* Значение, полученное от службы, для большинства служб может быть пустым.
status	*number*	*(optional)* Состояние службы: **0**, *DOBJECT_STATUS_UP* - служба доступна **1**, *DOBJECT_STATUS_DOWN* - служба недоступна
autoregistration	*array*	*(optional)* Массив объектов данных автоматической регистрации.
clock	*number*	Метка времени данных автоматической регистрации.
host	*string*	Имя узла сети.
ip	*string*	*(optional)* IP-адрес узла сети.
dns	*string*	*(optional)* Разрешенное DNS-имя по IP-адресу.
port	*string*	*(optional)* Порт узла сети.
host_metadata	*string*	*(optional)* Метаданные узла сети, отправленные агентом (на основе параметра конфигурации агента HostMetadata или HostMetadataItem).
tasks	*array*	*(optional)* Массив задач.
type	*number*	Тип задачи: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - результат удаленной команды
status	*number*	Статус выполнения удаленной команды: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - удаленная команда успешно выполнена **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - выполнение удаленной команды завершилось ошибкой
error	*string*	*(optional)* Сообщение об ошибке.
parent_taskid	*number*	ID родительской задачи.
more	*number*	*(optional)* 1 - есть еще данные истории для отправки
clock	*number*	*(optional)* Метка времени передачи данных (секунды).
ns	*number*	*(optional)* Метка времени передачи данных (наносекунды).
version	*string*	Версия прокси (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Информация об успешности запроса (’success’ или ’failed’).

name	value type	description
upload	*string*	Управление загрузкой исторических данных (history, autoregistration, host availability, network discovery). Возможные значения: **enabled** - нормальная работа **disabled** - сервер не принимает данные (возможно, из-за превышения внутреннего кэша)
tasks	*array*	*(optional)* Массив задач.
type	*number*	Тип задачи: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - удаленная команда
clock	*number*	Время создания задачи.
ttl	*number*	Время в секундах, после которого задача истекает.
commandtype	*number*	Тип удаленной команды: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - использовать пользовательский скрипт **1**, *ZBX_SCRIPT_TYPE_IPMI* - использовать IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - использовать SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - использовать Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - использовать глобальный скрипт (в настоящее время функционально эквивалентен пользовательскому скрипту)
command	*string*	Удаленная команда для выполнения.
execute_on	*number*	Цель выполнения для пользовательских скриптов: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - выполнить скрипт на агенте **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - выполнить скрипт на сервере **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - выполнить скрипт на прокси
port	*number*	*(optional)* Порт для команд Telnet и SSH.
authtype	*number*	*(optional)* Тип аутентификации для команд SSH.
username	*string*	*(optional)* Имя пользователя для команд Telnet и SSH.
password	*string*	*(optional)* Пароль для команд Telnet и SSH.
publickey	*string*	*(optional)* Открытый ключ для команд SSH.
privatekey	*string*	*(optional)* Закрытый ключ для команд SSH.
parent_taskid	*number*	ID родительской задачи.
hostid	*number*	ID целевого узла сети.

Пример:

proxy→server:


```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
```

```
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
```

```
"parent_taskid": 4,
"hostid": 10084
}]
}
```

Параметры конфигурации: Zabbix прокси

Обзор

Параметры конфигурации позволяют настраивать поведение прокси Zabbix.

Укажите значения параметров в файле конфигурации прокси (`zabbix_proxy.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его имя)
- Подробные сведения о параметрах

Параметр	Описание
AllowRoot	Разрешить запуск прокси от имени пользователя `root`.
AllowUnsupportedDBVersions	Разрешить прокси работать с неподдерживаемыми версиями базы данных.
APMListenIP	Указать IP-адрес, на котором сборщик APM должен ожидать входящие данные телеметрии.
APMListenPort	Указать порт, на котором сборщик APM должен ожидать входящие данные телеметрии.
APMTLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA), используемые для проверки сертификатов приложений или сборщиков OpenTelemetry, отправляющих данные сборщику APM.
APMTLSCertFile	Указать полный путь к файлу, содержащему сертификат или цепочку сертификатов сборщика APM.
APMTLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ сборщика APM.
CacheSize	Задать размер кэша конфигурации.
ConfigFrequency	Этот параметр **устарел** (используйте ProxyConfigFrequency). Задать частоту получения данных конфигурации от сервера Zabbix.
DataSenderFrequency	Задать частоту отправки собранных данных на сервер Zabbix.
DBHost	Указать узел сети базы данных (или каталог сокета).
DBName	Указать имя базы данных или путь к файлу базы данных для SQLite3.
DBPassword	Указать пароль базы данных.
DBPort	Указать порт узла сети базы данных Zabbix.
DBSchema	Указать имя схемы базы данных. Используется для PostgreSQL.
DBSocket	Указать путь к файлу сокета MySQL.
DBUser	Указать пользователя базы данных.
DBTLSConnect	Указать значение, принудительно устанавливающее TLS-соединение с базой данных.
DBTLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA), используемые для проверки сертификата базы данных.
DBTLSCertFile	Указать полный путь к файлу, содержащему сертификат прокси Zabbix для аутентификации в базе данных.
DBTLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.
DBTLSCipher	Указать список шифров, разрешённых прокси Zabbix для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.
DBTLSCipher13	Указать список наборов шифров, разрешённых прокси Zabbix для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.
DebugLevel	Задать уровень отладки.
EnableRemoteCommands	Разрешить удалённые команды от сервера Zabbix.
ExternalScripts	Указать расположение внешних скриптов.
Fping6Location	Указать расположение fping6.
FpingLocation	Указать расположение fping.
HistoryCacheSize	Задать размер кэша истории.
HistoryIndexCacheSize	Задать размер кэша индекса истории.
Hostname	Указать имя прокси.
HostnameItem	Указать элемент данных, используемый для задания Hostname (если он не определён).
HousekeepingFrequency	Задать частоту выполнения процедуры очистки (в часах).

Параметр	Описание
Include	Указать каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
JavaGateway	Указать IP-адрес (или имя узла сети) Java gateway Zabbix.
JavaGatewayPort	Указать порт Java gateway Zabbix.
ListenBacklog	Задать максимальное количество ожидающих подключений в очереди TCP.
ListenIP	Указать IP-адрес, на котором должен ожидать подключения траппер.
ListenPort	Указать порт, на котором должен ожидать подключения траппер.
LoadModule	Указать модуль, загружаемый при запуске прокси.
LoadModulePath	Указать полный путь к расположению модулей прокси.
LogFile	Указать файл журнала прокси.
LogFileSize	Задать максимальный размер файла журнала.
LogRemoteCommands	Включить запись выполненных команд оболочки в журнал с уровнем предупреждения.
LogSlowQueries	Задать максимальную продолжительность выполнения запроса к базе данных, после которой он записывается в журнал (в миллисекундах).
LogType	Указать тип вывода журнала.
MaxConcurrentChecksPerPoller	Задать максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым опросчиком HTTP-агента, опросчиком агента или опросчиком SNMP.
PidFile	Указать PID-файл.
ProxyBufferMode	Указать механизм хранения данных истории, обнаружения и автоматической регистрации (диск/память/гибридный режим).
ProxyConfigFrequency	Задать частоту получения данных конфигурации от сервера Zabbix в секундах.
ProxyLocalBuffer	Задать продолжительность хранения данных локально (в часах), даже если данные уже синхронизированы с сервером.
ProxyMemoryBufferAge	Задать максимальный возраст данных в буфере памяти прокси в секундах.
ProxyMemoryBufferSize	Задать размер кэша общей памяти для собранных данных истории, обнаружения и автоматической регистрации.
ProxyMode	Задать режим работы прокси (активный/пассивный).
ProxyOfflineBuffer	Задать продолжительность хранения данных (в часах) при отсутствии соединения с сервером Zabbix.
Server	Если для ProxyMode задано значение `active`: указать IP-адрес или DNS-имя сервера Zabbix (address:port) либо кластера (address:port;address2:port), откуда будут получаться данные конфигурации и куда будут отправляться данные. Если для ProxyMode задано значение `passive`: указать IP-адреса или DNS-имена сервера Zabbix.
SNMPTrapperFile	Указать временный файл, используемый для передачи данных от демона SNMP trap прокси.
SocketDir	Указать каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.
SourceIP	Указать исходный IP-адрес.
SSHKeyLocation	Указать расположение открытых и закрытых ключей для проверок и действий SSH.
SSLCertLocation	Указать расположение файлов сертификатов SSL-клиента для аутентификации клиента.
SSLKeyLocation	Указать расположение файлов закрытых ключей SSL для аутентификации клиента.
SSLCALocation	Указать расположение файлов центра сертификации (CA) для проверки сертификатов SSL-сервера.
StartAgentPollers	Задать количество предварительно разветвляемых экземпляров асинхронных опросчиков агента Zabbix.
StartAPMCollector	Задать значение 1, чтобы запустить сборщик APM.
StartBrowserPollers	Задать количество предварительно разветвляемых экземпляров опросчиков элементов данных браузера.
StartDBSyncers	Задать количество предварительно разветвляемых экземпляров синхронизаторов истории.
StartDiscoverers	Задать количество предварительно разветвляемых экземпляров обработчиков обнаружения.
StartHTTPAgentPollers	Задать количество предварительно разветвляемых экземпляров асинхронных опросчиков HTTP-агента.
StartHTTPPollers	Задать количество предварительно разветвляемых экземпляров опросчиков HTTP.
StartIPMIPollers	Задать количество предварительно разветвляемых экземпляров опросчиков IPMI.
StartJavaPollers	Задать количество предварительно разветвляемых экземпляров опросчиков Java.
StartODBCPollers	Задать количество предварительно разветвляемых экземпляров опросчиков ODBC.
StartPingers	Задать количество предварительно разветвляемых экземпляров пингеров ICMP.
StartPollersUnreachable	Задать количество предварительно разветвляемых экземпляров опросчиков для недоступных узлов сети (включая IPMI и Java).

Параметр	Описание
StartPollers	Задать количество предварительно разветвляемых экземпляров опросчиков.
StartPreprocessors	Задать количество предварительно запущенных экземпляров обработчиков предварительной обработки.
StartSNMPPollers	Задать количество предварительно разветвляемых экземпляров асинхронных опросчиков SNMP.
StartSNMPTrapper	Задать значение ”1”, чтобы запустить процесс траппера SNMP.
StartTelemetryQueryPollers	Задать количество предварительно разветвляемых экземпляров асинхронных опросчиков запросов телеметрии.
StartTrappers	Задать количество предварительно разветвляемых экземпляров трапперов.
StartVMwareCollectors	Задать количество предварительно разветвляемых экземпляров сборщика VMware.
StatsAllowedIP	Указать IP-адреса или DNS-имена внешних экземпляров Zabbix, которым разрешено запрашивать статистику.
TelemetryProvider	Задать источник данных телеметрии.
Timeout	Указать, как долго (в секундах) ожидать установления соединения и обмена данными с прокси Zabbix, агентом, веб-службой и выполнения проверок SNMP (за исключением элементов данных SNMP `walk[OID]` и `get[OID]`).
TLSAccept	Указать требуемый уровень шифрования для входящих подключений от сервера Zabbix.
TLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA), используемые для проверки сертификатов узлов сети при зашифрованном обмене данными между компонентами Zabbix.
TLSCertFile	Указать полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCipherAll	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Указать строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Указать строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Указать строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Указать требуемый уровень шифрования для подключений к серверу Zabbix.
TLSCRLFile	Указать полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного обмена данными между компонентами Zabbix.
TLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ прокси, используемый для зашифрованного обмена данными между компонентами Zabbix.
TLSListen	Требовать подключения только по TLS на сокете траппера.
TLSPSKFile	Указать полный путь к файлу, содержащему предварительно установленный ключ прокси, используемый для зашифрованного обмена данными с сервером Zabbix.
TLSPSKIdentity	Указать строку идентификатора предварительно установленного ключа, используемую для зашифрованного обмена данными с сервером Zabbix.
TLSServerCertIssuer	Указать разрешённого издателя сертификата сервера.
TLSServerCertSubject	Указать разрешённое назначение сертификата сервера.
TmpDir	Указать временный каталог.
TrapperTimeout	Указать время ожидания в секундах для: - получения данных конфигурации от сервера Zabbix; - выполнения глобального скрипта или удалённой команды.
UnavailableDelay	Задать частоту проверки доступности узла сети в период недоступности.
UnreachableDelay	Задать частоту проверки доступности узла сети в период недостижимости.
UnreachablePeriod	Задать, через сколько секунд недостижимости считать узел сети недоступным.
User	Понизить привилегии до уровня определённого существующего пользователя в системе.

Параметр	Описание
Vault	Указать поставщика хранилища.
VaultAppRoleID	Указать идентификатор AppRole хранилища HashiCorp.
VaultAppSecretID	Указать секретный идентификатор AppRole хранилища HashiCorp.
VaultDBPath	Указать расположение для получения учётных данных базы данных по ключам.
VaultPrefix	Указать пользовательский префикс для пути или запроса хранилища.
VaultTLSCertFile	Указать имя файла сертификата SSL, используемого для аутентификации клиента.
VaultTLSKeyFile	Указать имя файла закрытого ключа SSL, используемого для аутентификации клиента.
VaultToken	Указать токен аутентификации хранилища HashiCorp.
VaultURL	Указать URL сервера хранилища.
VMwareCacheSize	Задать размер общей памяти для хранения данных VMware.
VMwareFrequency	Задать частоту сбора данных от одной службы VMware в секундах.
VMwarePerfFrequency	Задать частоту получения статистики счётчиков производительности от одной службы VMware в секундах.
VMwareTimeout	Задать максимальное количество секунд, в течение которых `vmware collector` будет ожидать ответа от службы VMware.
WebDriverURL	Указать URL интерфейса WebDriver.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для демона, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

AllowRoot

Разрешить прокси запускаться от имени `root`. Если параметр отключен и прокси запущен пользователем `root`, прокси попытается переключиться на пользователя `zabbix`. Не имеет эффекта, если запуск выполнен от обычного пользователя.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

AllowUnsupportedDBVersions

Разрешает прокси работать с неподдерживаемыми версиями базы данных.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

APMListenIP

Укажите IP-адрес, на котором сборщик APM должен ожидать входящие данные телеметрии.
 Если параметр не указан, сборщик APM будет ожидать подключения на всех сетевых интерфейсах.

Значение по умолчанию: `0.0.0.0`

APMListenPort

Укажите порт, на котором сборщик APM должен ожидать входящие данные телеметрии.
 Порт по умолчанию — стандартный порт OpenTelemetry (OTLP/gRPC), который по умолчанию используется пакетами SDK и сборщиками OpenTelemetry.

Значение по умолчанию: 4317
 Диапазон: 1024-32767

APMTLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации для проверки сертификатов приложений или сборщиков OpenTelemetry, отправляющих данные в сборщик APM.
 Если этот параметр задан, сборщик APM принимает только зашифрованные соединения; в противном случае он принимает только незашифрованные соединения.

APMTLSCertFile

Укажите полный путь к файлу, содержащему сертификат или цепочку сертификатов сборщика APM.
 Этот параметр необходимо задать вместе с `APMTLSKeyFile` и только в том случае, если задан `APMTLSCAFile`.

APMTLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ сборщика APM.
 Этот параметр необходимо задать вместе с `APMTLSCertFile` и только в том случае, если задан `APMTLSCAFile`.

CacheSize

Задает размер кэша конфигурации в байтах. Размер разделяемой памяти для хранения данных узлов сети и элементов данных.

По умолчанию: 32M
 Диапазон: 128K-64G

ConfigFrequency

Этот параметр **устарел** (вместо него используйте ProxyConfigFrequency).
Задает частоту получения данных конфигурации от сервера Zabbix (в секундах).
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 3600
 Диапазон: 1-604800

DataSenderFrequency

Установите частоту отправки собранных данных с прокси на сервер Zabbix. Обратите внимание, что активный прокси по-прежнему будет опрашивать сервер Zabbix каждую секунду на наличие задач удаленных команд.
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 1
 Диапазон: 1-3600

DBHost

Укажите хост базы данных (или каталог сокета).

Для MySQL:

- localhost или пустая строка используют сокет UNIX-domain по умолчанию.

Для PostgreSQL:

- localhost разрешается через DNS (обычно в 127.0.0.1).
- Пустая строка использует сокет UNIX-domain по умолчанию.
- Путь (например, /var/run/pgbouncer) приводит к использованию UNIX-сокета по этому пути.
- Список, разделенный запятыми, может содержать несколько значений host:port. Для каждого хоста выполняются попытки по порядку, пока не будет установлено соединение с чтением и записью. Пример: DBHost=localhost:5431,127.0.0

По умолчанию: localhost

DBName

Укажите имя базы данных или путь к файлу базы данных для SQLite3 (многопроцессная архитектура Zabbix не позволяет использовать базу данных в памяти, например :memory:, file::memory:?cache=shared или file:memdb1?mode=memory&cache=
Не пытайтесь использовать ту же базу данных, которую использует сервер Zabbix.

Обязательный параметр: Да

DBPassword

Укажите пароль базы данных. Закомментируйте эту строку, если пароль не используется. Для SQLite игнорируется.

DBPort

Укажите порт узла сети базы данных Zabbix (см. DBHost). Для SQLite игнорируется.
 DBSocket и DBPort взаимоисключают друг друга в конфигурации прокси. Укажите только один из них или оставьте оба не заданными.

По умолчанию для MySQL: 3306

По умолчанию для PostgreSQL: 5432

Диапазон: 1024-65535

DBSchema

Укажите имя схемы базы данных. Используется для PostgreSQL.

DBSocket

Укажите путь к файлу сокета MySQL.
DBSocket и DBPort взаимоисключают друг друга в конфигурации прокси. Укажите только один из них или оставьте оба не заданными.

DBUser

Укажите пользователя базы данных. Игнорируется для SQLite.

DBTLSConnect

Укажите значение, чтобы принудительно использовать TLS-соединение с базой данных: - *required* - подключаться с использованием TLS; - *verify_ca* - подключаться с использованием TLS и проверять сертификат; - *verify_full* - подключаться с использованием TLS, проверять сертификат и проверять, что идентификатор базы данных, указанный в DBHost, соответствует ее сертификату
В MySQL, начиная с версии 5.7.11, и PostgreSQL поддерживаются следующие значения: ”required”, ”verify”, ”verify_full”.
В MariaDB, начиная с версии 10.2.6, поддерживаются значения ”required” и ”verify_full”.
По умолчанию не задано ни одно значение, и поведение зависит от конфигурации базы данных.

DBTLSCAFile

Укажите полное имя пути к файлу, содержащему сертификаты корневого CA для проверки сертификата базы данных.

Обязательно: нет (да, если DBTLSConnect установлен в *verify_ca* или *verify_full*)

DBTLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат прокси Zabbix для аутентификации в базе данных.

DBTLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.

DBTLSCipher

Укажите список шифров шифрования, которые прокси Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.

DBTLSCipher13

Укажите список наборов шифров шифрования, которые прокси Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

EnableRemoteCommands

Разрешить удаленные команды от сервера Zabbix.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

ExternalScripts

Укажите расположение внешних скриптов (зависит от переменной установки `datadir`, задаваемой на этапе компиляции).

По умолчанию: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Укажите расположение fping6. Убедитесь, что двоичный файл fping6 принадлежит root и для него установлен флаг SUID. Оставьте пустым (`Fping6Location=`), если ваша утилита `fping` умеет обрабатывать IPv6-адреса.

По умолчанию: `/usr/sbin/fping6`

FpingLocation

Укажите расположение fping. Убедитесь, что двоичный файл fping принадлежит root и для него установлен флаг SUID.

По умолчанию: `/usr/sbin/fping`

HistoryCacheSize

Установите размер кэша истории в байтах. Размер разделяемой памяти для хранения данных истории.

По умолчанию: `16M`
 Диапазон: 128K-16G

HistoryIndexCacheSize

Установите размер кэша индекса истории в байтах. Размер общей памяти для индексирования данных истории, хранящихся в кэше истории. Размер кэша индекса составляет примерно 100 байт на один элемент данных.

По умолчанию: `4M`
 Диапазон: 128K-16G

Hostname

Укажите уникальное имя прокси с учетом регистра. Убедитесь, что имя прокси известно серверу.
Допустимые символы: буквенно-цифровые, '.', ' ', '_' и '-'. Максимальная длина: 128

По умолчанию: задается HostnameItem

HostnameItem

Укажите элемент данных для задания `Hostname` (если не определено).
 Этот элемент данных будет выполняться на прокси так же, как и на агенте.
 Игнорируется, если задан `Hostname`.
Не поддерживает пользовательские параметры, счетчики производительности или псевдонимы; поддерживает `system.run[]`.

По умолчанию: `system.hostname`

HousekeepingFrequency

Задайте частоту выполнения процедуры очистки (в часах). Очистка — это удаление устаревшей информации из базы данных.
Примечание: чтобы снизить нагрузку при запуске прокси, очистка откладывается на 30 минут после запуска прокси. Таким образом, если `HousekeepingFrequency` равен 1, первая процедура очистки после запуска прокси будет выполнена через 30 минут, а затем будет повторяться каждый час.
Автоматическую очистку можно отключить, установив для HousekeepingFrequency значение 0. В этом случае процедуру очистки можно запустить только с помощью параметра управления во время выполнения `housekeeper_execute`.
См. также параметры управления во время выполнения и сведения о процедуре очистки.

По умолчанию: 1
 Диапазон: 0-24

Include

Укажите каталог или отдельные файлы, которые нужно включить в файл конфигурации.
 Во время установки Zabbix этот параметр устанавливается в `/usr/local/etc`, если вы не измените его на этапе компиляции.

Общие правила для параметра `Include`:

- Путь может быть относительным к расположению файла `zabbix_proxy.conf`.
- В одном файле конфигурации можно указать несколько параметров `Include`.
- Zabbix обрабатывает параметр `Include` сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила для включения каталогов или шаблонов файлов:

- Не следует предполагать какой-либо порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общее значение более конкретным), поскольку невозможно предсказать, какое значение будет использовано.
- Обратите внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан файл `include/my_specific.conf.BAK`), поскольку будут включены оба файла.

Чтобы включать только нужные файлы из каталога, используйте подстановочный знак (`*`) для сопоставления с шаблоном.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Укажите IP-адрес или имя хоста Zabbix Java gateway. Требуется только в том случае, если запущены опросчики Java.

JavaGatewayPort

Укажите порт Zabbix Java gateway.

По умолчанию: 10052
 Диапазон: 1024-32767

ListenBacklog

Установите максимальное количество ожидающих соединений в очереди TCP.
Значение по умолчанию — жестко заданная константа, которая зависит от системы.
Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть незаметно усечены до 'implementation-specified maximum'.

По умолчанию: `SOMAXCONN`
 Диапазон: 0 - INT_MAX

ListenIP

Укажите IP-адрес для прослушивания trapper. Список, разделенный запятыми.
Trapper будет прослушивать все сетевые интерфейсы, если этот параметр не указан.

По умолчанию: `0.0.0.0`

ListenPort

Укажите порт прослушивания trapper.

По умолчанию: 10051
 Диапазон: 1024-32767

LoadModule

Укажите модуль, который необходимо загрузить при запуске прокси. Модули используются для расширения функциональности прокси. Модуль должен находиться в каталоге, указанном в `LoadModulePath`, либо перед именем модуля необходимо указать путь. Если указанный путь является абсолютным (начинается с /), параметр `LoadModulePath` игнорируется.
 Форматы: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Можно указать несколько параметров `LoadModule`.

LoadModulePath

Полный путь к расположению модулей прокси. Значение по умолчанию зависит от параметров компиляции.

LogFile

Укажите файл журнала прокси.

Обязательно: Да, если `LogType` имеет значение `file`; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: Если достигается предел размера файла журнала и ротация файлов по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включить запись выполненных команд оболочки в журнал как предупреждения.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

LogSlowQueries

Установите максимальную длительность выполнения запроса к базе данных, по истечении которой он будет записан в журнал (в миллисекундах).
0 - не записывать медленные запросы.
Эта опция становится доступной начиная с `DebugLevel=3`.

По умолчанию: 0
 Диапазон: 0-3600000

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром `LogFile`; - *system* - записывать журнал в syslog; - *console* - записывать журнал в стандартный вывод.

По умолчанию: `file`

MaxConcurrentChecksPerPoller

Установите максимальное количество асинхронных проверок, которые могут выполняться одновременно каждым опрашивателем HTTP-агента, опрашивателем агента или SNMP-опрашивателем. См. `StartHTTPAgentPollers`, `StartAgentPollers` и `StartSNMPPollers`.

По умолчанию: 1000
 Диапазон: 1-1000

PidFile

Укажите PID-файл.

По умолчанию: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Укажите механизм хранения данных истории, обнаружения и автоматической регистрации: - *disk* - данные хранятся в базе данных и загружаются из базы данных; - *memory* - данные хранятся в памяти и загружаются из памяти. Если буфер исчерпает память, старые данные будут отброшены. При завершении работы буфер будет удален. - *hybrid* - буфер прокси обычно работает как в режиме памяти, пока не исчерпает память или пока самая старая запись не превысит заданный

возраст. Если это происходит, буфер сбрасывается в базу данных и работает как в режиме диска, пока все данные не будут загружены, после чего снова начинает работать с памятью. При завершении работы буфер памяти сбрасывается в базу данных.

См. также: Буфер памяти прокси.

По умолчанию: disk
 Значения: disk; memory; hybrid

ProxyConfigFrequency

Установите частоту получения данных конфигурации от сервера Zabbix в секундах.
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 10
 Диапазон: 1-604800

ProxyLocalBuffer

Установите длительность локального хранения данных (в часах), даже если данные уже были синхронизированы с сервером..
Этот параметр можно использовать, если локальные данные будут использоваться сторонними приложениями.

По умолчанию: 0
 Диапазон: 0-720

ProxyMemoryBufferAge

Установите максимальный возраст данных в буфере памяти прокси в секундах.
 Когда параметр включен (не равен нулю) и записи в буфере памяти прокси старше указанного значения, это принудительно переводит буфер прокси в режим базы данных, пока все записи не будут загружены на сервер. Этот параметр должен быть меньше или равен параметру ProxyOfflineBuffer.

По умолчанию: 0
 Диапазон: 0;600-864000

ProxyMemoryBufferSize

Установите размер кэша общей памяти для собранных данных истории, обнаружения и автоматической регистрации (в байтах). Если включено (не равно нулю), прокси будет хранить данные истории, обнаружения и автоматической регистрации в памяти, пока кэш не заполнится или пока сохраненные записи не станут старше заданного ProxyMemoryBufferAge. Этот параметр нельзя использовать вместе с параметром ProxyLocalBuffer.

По умолчанию: 0
 Диапазон: 0;128K-2G

ProxyMode

Установите режим работы прокси.
0 - прокси в активном режиме
1 - прокси в пассивном режиме
Примечание: конфигурационные данные прокси (в том числе конфиденциальные) могут стать доступны лицам, имеющим доступ к порту trapper сервера Zabbix, при использовании активного прокси. Это возможно, поскольку любой может выдать себя за активный прокси и запросить конфигурационные данные; аутентификация при этом не выполняется.

По умолчанию: 0
 Диапазон: 0-1

ProxyOfflineBuffer

Установите длительность хранения данных (в часах) в случае отсутствия связи с сервером Zabbix.
Более старые данные будут потеряны.

По умолчанию: 1
 Диапазон: 1-720

Сервер

Если для ProxyMode установлено значение active: укажите IP-адрес или DNS-имя сервера Zabbix (address:port) или кластер (address:port;address2:port), откуда получать данные конфигурации и куда отправлять данные.
Если порт не указан, используется порт по умолчанию.
Узлы кластера должны быть разделены точкой с запятой.

Если для ProxyMode установлено значение passive: укажите IP-адреса, при необходимости в нотации CIDR, или DNS-имена сервера Zabbix. Список, разделенный запятыми.
 Входящие соединения будут приниматься только с адресов, перечисленных здесь. Если поддержка IPv6 включена, то 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 обрабатываются одинаково.
::/0 позволит использовать любой IPv4- или IPv6-адрес. 0.0.0.0/0 можно использовать, чтобы разрешить любой IPv4-адрес.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательно: да

SNMPTrapperFile

Укажите временный файл, используемый для передачи данных от демона SNMP trap к прокси.
Должен совпадать с параметром в `zabbix_trap_receiver.pl` или в файле конфигурации SNMPTT.

По умолчанию: `/tmp/zabbix_traps.tmp`

SocketDir

Укажите каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.

По умолчанию: `/tmp`

SourceIP

Укажите исходящий IP-адрес для:

- исходящих соединений с сервером Zabbix
- соединений без использования агента (VMware, SSH, JMX, SNMP, Telnet и простых проверок)
- соединений элемента данных HTTP agent
- HTTP-запросов JavaScript в элементе данных script
- HTTP-запросов JavaScript при предварительной обработке
- соединений с Vault

SSHKeyLocation

Укажите расположение открытых и закрытых ключей для SSH-проверок и действий.

SSLCertLocation

Укажите расположение файлов сертификата SSL-клиента для аутентификации клиента.
Этот параметр используется в веб-мониторинге, для связи с Vault и источником телеметрических данных ClickHouse.

SSLKeyLocation

Укажите расположение файлов закрытых ключей SSL для аутентификации клиента.
Этот параметр используется в веб-мониторинге, для связи с Vault и для источника телеметрических данных ClickHouse.

SSLCALocation

Укажите расположение файлов центра сертификации (CA) для проверки сертификата SSL-сервера.
Обратите внимание, что значение этого параметра будет задано для параметра `CURLOPT_CAPATH` библиотеки libcurl. Для версий libcurl до 7.42.0 этот параметр действует только в том случае, если libcurl скомпилирована для использования OpenSSL. Дополнительную информацию см. на веб-странице cURL.
Этот параметр используется в веб-мониторинге, аутентификации SMTP, элементах данных HTTP agent, при обмене данными с Vault и для источника телеметрических данных ClickHouse.

StartAgentPollers

Задает количество предварительно порожденных экземпляров `pollers` агента Zabbix. См. `MaxConcurrentChecksPerPoller`.

По умолчанию: 1
 Диапазон: 0-1000

StartAPMCollector

Установите значение 1, чтобы запустить предварительно разветвленный экземпляр сборщика APM (процессы `apm manager` и `apm worker`).

Значение по умолчанию: 0
 Диапазон: 0-1

StartBrowserPollers

Установите количество предварительно порожденных экземпляров `pollers` для элементов данных браузера.

По умолчанию: 1
 Диапазон: 0-1000

StartDBSyncers

Установите количество предварительно созданных экземпляров `history syncers`.
Примечание: Будьте осторожны при изменении этого значения: его увеличение может принести больше вреда, чем пользы.

По умолчанию: 4
 Диапазон: 1-100

StartDiscoverers

Установите количество предварительно порожденных экземпляров `discovery workers`.

По умолчанию: 5
 Диапазон: 0-1000

StartHTTPAgentPollers

Задает количество предварительно созданных экземпляров HTTP agent pollers. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartHTTPPollers

Установите количество предварительно порожденных экземпляров HTTP pollers.

По умолчанию: 1
 Диапазон: 0-1000

StartIPMIPollers

Установите количество предварительно созданных экземпляров IPMI pollers.

По умолчанию: 0
 Диапазон: 0-1000

StartJavaPollers

Установите количество предварительно порожденных экземпляров Java pollers.

По умолчанию: 0
 Диапазон: 0-1000

StartODBCPollers

Установите количество предварительно созданных экземпляров ODBC pollers.

По умолчанию: 1
 Диапазон: 0-1000

StartPingers

Установите количество предварительно порожденных экземпляров ICMP pingers.

По умолчанию: 1
 Диапазон: 0-1000

StartPollersUnreachable

Задает количество предварительно порожденных экземпляров pollers for unreachable hosts (включая IPMI и Java). Если запущены обычные pollers, IPMI или Java pollers, должен быть запущен как минимум один poller for unreachable hosts.

По умолчанию: 1
 Диапазон: 0-1000

StartPollers

Установите количество предварительно созданных экземпляров pollers.

По умолчанию: 5
 Диапазон: 0-1000

StartPreprocessors

Установите число предварительно запущенных экземпляров потоков предварительной обработки worker не меньше количества доступных ядер CPU. Следует указать большее число worker, если предварительная обработка не ограничена CPU и выполняет много сетевых запросов.

По умолчанию: 16
 Диапазон: 1-1000

StartSNMPPollers

Задает количество предварительно созданных экземпляров SNMP pollers. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartTelemetryQueryPollers

Задайте количество предварительно запущенных экземпляров асинхронных опросчиков запросов Telemetry. См. MaxConcurrentChecksPerPoller.

Значение по умолчанию: 1
 Диапазон: 0-1000

StartSNMPTrapper

Установите значение "1", чтобы запустить процесс SNMP trapper.

По умолчанию: 0
 Диапазон: 0-1

StartTrappers

Задает количество предварительно созданных экземпляров trappers.
Trappers принимают входящие соединения от Zabbix sender и активных агентов.

По умолчанию: 5
 Диапазон: 0-1000

StartVMwareCollectors

Установите количество предварительно порожденных экземпляров `vmware collector`.

По умолчанию: 0
 Диапазон: 0-250

StatsAllowedIP

Укажите IP-адреса (при необходимости в нотации CIDR) или DNS-имена внешних экземпляров Zabbix, которым разрешено запрашивать статистику. Список, разделенный запятыми.
 Если этот параметр не задан, запросы статистики приниматься не будут.
Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются эквивалентными, а `::/0` разрешит любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес.

Пример:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Задайте источник данных телеметрии в формате `<provider>[;<options>]`, где:

- `<provider>` - один из поддерживаемых источников данных телеметрии: `clickhouse`
- `<options>` - пары `key=value`, разделённые запятыми. Значения можно заключать в кавычки.

Параметры ClickHouse:

- `url` - задаёт URL HTTP или HTTPS, используемый для подключения к источнику данных телеметрии.
- `db` - задаёт имя базы данных.
- `username` - задаёт имя пользователя, используемое для входа в базу данных.
- `password` - задаёт пароль, используемый для входа в базу данных.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - задают параметры защищённого (TLS/SSL) подключения к базе данных.
- `source_ip` - задаёт исходный IP-адрес, используемый для исходящих подключений к источнику данных телеметрии.
- `vault_path` - задаёт путь к секрету Vault, содержащему значения `username` и `password`. Не используйте вместе с параметрами `username` и `password`.

Пример для ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Пример для ClickHouse с Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Таймаут

Укажите, сколько времени ожидать (в секундах) при установлении соединения и обмене данными с сервером Zabbix, агентом, веб-сервисом и устаревшими проверками SNMP (одиночный OID-номер или строка).

Этот параметр определяет длительность для различных операций связи:

- Выполнение удаленной команды на агенте Zabbix
- Выполнение команды SSH / Telnet
- Перепланирование элементов данных, когда интерфейс IPMI становится недоступным
- Отправка ответа серверу Zabbix при сбое обмена данными из-за проблем с правами доступа или шифрованием
- Срок для асинхронных сокетов IPC и параметров управления во время выполнения
- DNS-запросы асинхронного poller
- Ответ на heartbeat активной проверки
- Получение данных (значений) агента Zabbix от активных агентов
- Получение данных от Zabbix sender
- Отправка списка активных проверок агенту Zabbix

Этот таймаут **не** будет использоваться для тех проверок, у которых в веб-интерфейсе настроены параметры гибкого таймаута (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных). Например, элементы данных SNMP `walk[OID]` и `get[OID]` используют настроенный таймаут из веб-интерфейса; устаревшие проверки SNMP по-прежнему используют значение таймаута сервера.

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Укажите уровень шифрования, требуемый для входящих подключений от сервера Zabbix. Используется для пассивного прокси, игнорируется на активном прокси. Можно указать несколько значений, разделяя их запятыми: - *unencrypted* -

принимать подключения без шифрования (по умолчанию); - *psk* - принимать подключения с TLS и предварительно общим ключом (PSK); - *cert* - принимать подключения с TLS и сертификатом.

Обязательно: да для пассивного прокси, если заданы параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата узла-пира, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Укажите полный путь к файлу, содержащему сертификат прокси или цепочку сертификатов, используемых для зашифрованной связи между компонентами Zabbix.

TLSCipherAll

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

TLSCipherPSK

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Укажите уровень шифрования, требуемый для соединений прокси с сервером Zabbix. Используется для активного прокси, игнорируется для пассивного прокси. Можно указать только одно значение: - *unencrypted* - подключение без шифрования (по умолчанию); - *psk* - подключение с использованием TLS и предварительно общего ключа (PSK); - *cert* - подключение с использованием TLS и сертификата.

Обязательно: да для активного прокси, если определены параметры TLS-сертификата или PSK (даже для соединения *unencrypted*); в противном случае нет

TLSCRLFile

Укажите полное имя пути к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ прокси, используемый для зашифрованной связи между компонентами Zabbix.

TLSListen

Требовать подключения только по TLS на сокете trapper.

Поддерживаемые значения:

- required - принимать только TLS-подключения

TLSPSKFile

Укажите полный путь к файлу, содержащему предварительно общий ключ прокси, используемый для зашифрованного обмена данными с сервером Zabbix.

TLSPSKIdentity

Укажите строку идентификатора предварительно общего ключа, используемую для зашифрованной связи с сервером Zabbix.

TLSServerCertIssuer

Укажите допустимого издателя серверного сертификата.

TLSServerCertSubject

Укажите допустимый subject сертификата сервера.

TmpDir

Укажите временный каталог.

По умолчанию: `/tmp`

TrapperTimeout

Укажите тайм-аут в секундах для:
 - получения данных конфигурации с сервера Zabbix;
 - выполнения глобального скрипта или удаленной команды.

По умолчанию: 300
 Диапазон: 1-300

UnavailableDelay

Установите частоту проверки доступности узла сети в течение периода недоступности в секундах.

По умолчанию: 60
 Диапазон: 1-3600

UnreachableDelay

Установите частоту проверки доступности узла сети в течение периода недоступности в секундах.

По умолчанию: 15
 Диапазон: 1-3600

UnreachablePeriod

Указывает, через сколько секунд недоступности считать узел сети недоступным.

По умолчанию: 45
 Диапазон: 1-3600

Пользователь

Снижает привилегии до указанного существующего пользователя в системе.
Имеет эффект только при запуске от имени `root`, если `AllowRoot` отключен.

По умолчанию: `zabbix`

Vault

Укажите поставщика хранилища: - *HashiCorp* - HashiCorp KV Secrets Engine version 2; - *CyberArk* - CyberArk Central Credential Provider
Должен соответствовать поставщику хранилища, заданному в веб-интерфейсе.

По умолчанию: `HashiCorp`

VaultAppRoleID

Укажите AppRole ID HashiCorp Vault.
 Если задано, включает аутентификацию AppRole. Не может использоваться вместе с VaultToken.
 Поддерживается только для HashiCorp Vault.

VaultAppSecretID

Укажите secret ID AppRole в HashiCorp Vault.
 Поддерживается только для HashiCorp Vault.

VaultDBPath

Укажите расположение для получения учетных данных базы данных по ключам. Это путь или запрос Vault, в зависимости от используемого Vault.

Ключи, используемые для HashiCorp, — `password` и `username`.

Пример пути с `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Пример пути без `VaultPrefix`:

```
secret/zabbix/database
```

Ключи, используемые для CyberArk, — `Content` и `UserName`.

Пример:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Этот параметр можно использовать только если не указаны `DBUser` и `DBPassword`.

VaultPrefix

Укажите пользовательский префикс для пути или запроса Vault в зависимости от используемого Vault. Если не указано, будут использованы наиболее подходящие значения по умолчанию.
Обратите внимание, что 'data' автоматически добавляется после точки монтирования для HashiCorp, если `VaultPrefix` не указан.
Обратите внимание, что 'data' автоматически добавляется после точки монтирования для HashiCorp, если `VaultPrefix` не указан.

Пример префикса для HashiCorp:

```
/v1/secret/data/zabbix/
```

Пример префикса для Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Укажите имя файла SSL-сертификата, используемого для аутентификации клиента. Файл сертификата должен быть в формате PEM1.
Если файл сертификата также содержит закрытый ключ, оставьте поле файла SSL-ключа пустым.
Каталог, содержащий этот файл, задается параметром конфигурации SSLCertLocation.
Этот параметр можно не указывать, но для хранилища CyberArkCCP его рекомендуется использовать.

VaultTLSKeyFile

Укажите имя файла закрытого ключа SSL, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM1.
Каталог, содержащий этот файл, задается параметром конфигурации SSLKeyLocation.
Этот параметр можно не указывать, но для хранилища CyberArkCCP его использование рекомендуется.

VaultToken

Укажите токен аутентификации HashiCorp Vault, который должен быть сгенерирован исключительно для прокси Zabbix с правом только на чтение к пути, указанному в необязательном параметре конфигурации VaultDBPath.
Это ошибка, если `VaultToken` и переменная среды `VAULT_TOKEN` заданы одновременно.
 Не может использоваться вместе с VaultAppRoleID.
 Поддерживается только для HashiCorp Vault.

VaultURL

Укажите URL сервера vault. Будет использоваться системный каталог сертификатов CA, если не указан SSLCALocation.

По умолчанию: `https://127.0.0.1:8200`

VMwareCacheSize

Установите размер разделяемой памяти для хранения данных VMware.
Для мониторинга использования кэша VMware можно использовать внутреннюю проверку `zabbix[vmware,buffer,...]` (см. Внутренние проверки).
Обратите внимание, что разделяемая память не выделяется, если не настроены экземпляры сборщика vmware для запуска.

По умолчанию: 8M
 Диапазон: 256K-2G

VMwareFrequency

Установите частоту сбора данных с одного VMware-сервиса (в секундах).
Эта частота должна быть установлена равной наименьшему интервалу обновления любого элемента данных мониторинга VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwarePerfFrequency

Установите частоту получения статистики счетчиков производительности от одного VMware service (в секундах).
Эта частота должна быть установлена равной наименьшему интервалу обновления любого элемента данных VMware monitoring item, который использует счетчики производительности VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwareTimeout

Установите максимальное количество секунд, в течение которых `vmware collector` будет ожидать ответа от службы VMware (vCenter или гипервизора ESX).

По умолчанию: 10
 Диапазон: 1-300

WebDriverURL

Укажите URL интерфейса WebDriver.

Пример (используется с автономным сервером Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

Агент

Обзор Агент Zabbix устанавливается на контролируемом узле сети для активного мониторинга локальных ресурсов и приложений (жестких дисков, памяти, статистики процессора и т. д.).

Агент локально собирает эксплуатационную информацию и передает данные серверу Zabbix для дальнейшей обработки. В случае сбоев (например, при заполнении жесткого диска или аварийном завершении процесса службы) сервер Zabbix может активно уведомить администраторов соответствующей машины, сообщившей о сбое.

Агенты Zabbix отличаются высокой эффективностью благодаря использованию собственных системных вызовов для сбора статистической информации.

На одном узле сети можно запустить несколько экземпляров агента.

См. также:

- Протоколы пассивных и активных проверок
- Запуск агента от имени пользователя root
- Параметры конфигурации: агент Zabbix
- Параметры конфигурации: агент Zabbix (Windows)
- Агент Zabbix в Microsoft Windows
- Агент Zabbix в IBM AIX

Пассивные и активные проверки

Агенты Zabbix могут выполнять пассивные и активные проверки:

- Пассивные проверки - агент Zabbix отвечает на запрос от сервера Zabbix (или прокси). Например, сервер запрашивает данные (например, загрузку CPU), а агент возвращает результат.

- Активные проверки - агент Zabbix собирает и отправляет данные, не дожидаясь запроса от сервера Zabbix (или прокси). Сначала он получает список элементов мониторинга от сервера (загрузка CPU, доступная память и т. д.), затем собирает необходимые данные и периодически отправляет ему новые значения.

Тип проверки агента настраивается путем выбора соответствующего типа элемента данных. Агент Zabbix обрабатывает элементы данных типа *Zabbix agent* или *Zabbix agent (active)*.

Поддерживаемые платформы Список поддерживаемых платформ см. на странице Требования.

Агент в Unix-подобных системах Агент Zabbix в Unix-подобных системах запускается на узле сети, который находится под мониторингом.

Установка

Агент Zabbix можно установить в системах на базе Linux одним из следующих способов:

- Пакеты Zabbix - выберите компонент Agent (после выбора версии Zabbix, дистрибутива ОС и версии ОС) и следуйте предоставленным инструкциям;
- Исходные файлы Zabbix - загрузите исходные файлы и скомпилируйте агент Zabbix, настроив исходные файлы с параметром `--enable-agent`.

Attention:
В целом 32-разрядные агенты Zabbix будут работать в 64-разрядных системах, но в некоторых случаях могут возникать сбои.

Note:
Предварительно скомпилированные двоичные файлы агента Zabbix доступны для загрузки для macOS, IBM AIX, FreeBSD, OpenBSD и Solaris. Устаревшие двоичные файлы, совместимые с текущей версией сервера/прокси Zabbix, доступны для NetBSD и HP-UX. Инструкции по установке и запуску агента в IBM AIX см. в разделе Агент Zabbix в IBM AIX.

При установке из пакета

Zabbix агент работает как демон. Агента можно запустить, выполнив:

```
systemctl start zabbix-agent
```

Эта команда будет работать на большинстве систем GNU/Linux. На других системах вам, возможно, потребуется выполнить:

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

Для остановки/перезапуска/просмотра состояния используйте следующие команды:

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

Если установка выполнена из двоичного архива

После распаковки архива найдите двоичный файл `zabbix_agentd` и файл конфигурации `zabbix_agentd.conf`. Их расположение внутри архива зависит от платформы (например, `sbin/` и `conf/` или `usr/sbin/` и `etc/zabbix/`). Запустите двоичный файл непосредственно из распакованного каталога и укажите файл конфигурации с помощью параметра `-c`. Например:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

В UNIX-подобных системах агент предназначен для работы от имени пользователя без прав root. Если агент запущен от имени root, он переключается на пользователя `zabbix` (или пользователя, указанного параметром [User](<PARAMS>#user)), если в файле конфигурации не задано [AllowRoot=1](<PARAMS>#allowroot). Этот пользователь должен существовать в системе; в противном случае агент не запускается.

В IBM AIX агент также можно запускать с помощью System Resource Controller (SRC) и настроить на запуск при загрузке системы; см. раздел Агент Zabbix в IBM AIX.

Запуск вручную

Вы можете запустить агент Zabbix, непосредственно выполнив двоичный файл `zabbix_agentd`, например:

```
zabbix_agentd
```

Это работает, если каталог, содержащий `zabbix_agentd`, включен в переменную `PATH`.

Агент в системах Windows Агент Zabbix в Windows работает как служба Windows.

Установка

Агент Zabbix можно установить в Windows одним из следующих способов:

- Предварительно скомпилированные двоичные файлы агента Zabbix - загрузите MSI-пакет установщика агента Zabbix и следуйте инструкциям на странице Установка агента Windows из MSI;
- Исходные тексты Zabbix - загрузите исходные файлы и следуйте инструкциям на странице Сборка агента Zabbix в Windows.

Дополнительные сведения об установке агента Zabbix (из ZIP-архива) в качестве службы Windows см. на странице Агент Zabbix в Microsoft Windows.

Параметры С агентом Zabbix можно использовать следующие параметры командной строки:

Параметр	Описание
Агент UNIX и Windows	
`-c --config <config-file>`	Путь к файлу конфигурации. Этот параметр можно использовать для указания файла конфигурации, отличного от используемого по умолчанию.
`-f --foreground`	Запустить агент Zabbix в режиме переднего плана (по умолчанию: `true`).
`-p --print`	Вывести известные элементы данных и завершить работу. Обратите внимание: чтобы также получить результаты пользовательских параметров, необходимо указать файл конфигурации (если он находится не в расположении по умолчанию).
`-t --test <item key>`	Проверить указанный элемент данных и завершить работу. Обратите внимание: чтобы также получить результаты пользовательских параметров, необходимо указать файл конфигурации (если он находится не в расположении по умолчанию).
`-T --test-config`	Проверить файл конфигурации и завершить работу.
`-h --help`	Отобразить справочную информацию.
`-V --version`	Отобразить номер версии.
Только агент UNIX	
`-R --runtime-control <option>`	Выполнить административные функции. См. раздел управление во время выполнения.
Только агент Windows	
`-m --multiple-agents`	Использовать несколько экземпляров агента (с параметрами `-i`, `-d`, `-s`, `-x`). Чтобы различать имена служб экземпляров, имя каждой службы будет включать значение Hostname из указанного файла конфигурации.
`-S --startup-type <value>`	Установить тип запуска службы агента Zabbix для Windows. Допустимые значения: `automatic` - *(по умолчанию)* автоматически запускать службу при запуске Windows; `delayed` - отложить запуск службы до завершения запуска служб, запущенных автоматически (доступно в Windows Server 2008/Vista и более поздних версиях); `manual` - запускать службу вручную (пользователем или приложением); `disabled` - отключить службу, чтобы ее нельзя было запустить пользователю или приложению. Этот параметр можно использовать вместе с параметром `-i` или отдельно для изменения типа запуска уже установленной службы.
`-i --install`	Установить агент Zabbix для Windows как службу.
`-d --uninstall`	Удалить службу агента Zabbix для Windows.
`-s --start`	Запустить службу агента Zabbix для Windows.
`-x --stop`	Остановить службу агента Zabbix для Windows.

Примеры использования параметров командной строки:

- Вывод всех встроенных элементов данных агента со значениями.

- Проверка пользовательского параметра с ключом `mysql.ping`, определенным в указанном файле конфигурации.
- Установка службы агента Zabbix для Windows с использованием пути по умолчанию к файлу конфигурации.
- Установка службы агента Zabbix [Hostname] для Windows с использованием файла конфигурации, расположенного в той же папке, что и исполняемый файл агента, и создание уникального имени службы путем добавления значения `Hostname` из файла конфигурации.
- Изменение типа запуска установленной службы агента Zabbix для Windows с использованием файла конфигурации, расположенного в той же папке, что и исполняемый файл агента.

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

Управление во время выполнения

С помощью параметров управления во время выполнения можно изменять уровень журнала процессов агента.

Параметр	Описание	Цель
`log_level_increase[=<target>]`	Повысить уровень журнала. Если цель не указана, затрагиваются все процессы.	Цель можно указать следующим образом: **тип процесса** — все процессы указанного типа (например, `listener`) См. все типы процессов агента. **тип процесса,N** — тип и номер процесса (например, `listener,3`) **pid** — идентификатор процесса (от 1 до 65535). Для больших значений укажите цель в формате «тип процесса,N».
`log_level_decrease[=<target>]`	Понизить уровень журнала. Если цель не указана, затрагиваются все процессы.	
`userparameter_reload`	Перезагрузить значения параметров `UserParameter` и `Include` из текущего файла конфигурации.	

Примеры:

- Повышение уровня журнала всех процессов
- Повышение уровня журнала третьего процесса `listener`
- Повышение уровня журнала процесса с PID 1234
- Понижение уровня журнала всех процессов активных проверок

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

> **Note:**
> Управление во время выполнения не поддерживается в OpenBSD, NetBSD и Windows.

Типы процессов агента Типы процессов агента следующие:

- `active checks` - процесс для выполнения активных проверок;
- `collector` - процесс для сбора данных;
- `listener` - процесс для прослушивания пассивных проверок.

Файл журнала агента можно использовать для наблюдения за этими типами процессов.

Файл журнала агента создается с правами на чтение и запись только для владельца файла. Кроме того, файл доступен для чтения группе владельца. Все остальные права запрещены.

Запуск от имени пользователя Агент Zabbix в UNIX предназначен для запуска от имени пользователя, не являющегося пользователем root. Он будет работать от имени того пользователя, не являющегося пользователем root, от имени которого был запущен. Таким образом, вы можете запускать агент от имени любого пользователя, не являющегося пользователем root, без каких-либо проблем.

Если вы попытаетесь запустить его от имени пользователя «root», он переключится на жестко заданного пользователя «zabbix», который должен присутствовать в вашей системе. Вы можете запускать агент от имени пользователя «root» только после соответствующего изменения параметра `AllowRoot` в файле конфигурации агента.

Файл конфигурации Путь к файлу конфигурации по умолчанию зависит от операционной системы и способа установки Zabbix:

- В UNIX по умолчанию используется:
  - При установке из исходных кодов: `/usr/local/etc/zabbix_agentd.conf` или путь, заданный переменными `--sysconfdir` или `--prefix` на этапе компиляции.
  - При установке из пакетов и из appliance: `/etc/zabbix/zabbix_agentd.conf`.
- В Windows по умолчанию используется `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Если на узле сети запущен один экземпляр агента, он может использовать файл конфигурации по умолчанию или файл конфигурации, указанный в командной строке. При использовании нескольких экземпляров каждый экземпляр агента должен иметь собственный файл конфигурации (один из экземпляров может использовать файл конфигурации по умолчанию).

Подробнее о настройке агента Zabbix см. параметры файла конфигурации для агента Zabbix в Unix или агента Zabbix в Windows.

Локаль Обратите внимание, что агенту требуется локаль UTF-8, чтобы некоторые текстовые элементы данных агента могли возвращать ожидаемое содержимое. На большинстве современных Unix-подобных систем локаль UTF-8 используется по умолчанию, однако на некоторых системах ее может потребоваться задать явно.

Код выхода Агент Zabbix возвращает 0 в случае успешного завершения и 1 в случае ошибки.

Протоколы пассивных и активных проверок

Этот раздел содержит сведения о протоколе для пассивных и активных проверок, выполняемых Zabbix agent и Zabbix agent 2.

Zabbix использует протокол обмена данными на основе JSON для взаимодействия с агентами.

Протоколы Zabbix agent и Zabbix agent 2 были унифицированы начиная с Zabbix 7.0. Различие между запросами/ответами Zabbix agent и Zabbix agent 2 выражается значением тега ”variant”.

Пассивные проверки

Пассивная проверка — это простой запрос данных. Сервер или прокси Zabbix запрашивает некоторые данные (например, загрузку CPU), а агент Zabbix отправляет результат обратно на сервер.

Пассивные проверки выполняются асинхронно — не требуется получать ответ на один запрос до начала других проверок. Разрешение DNS также выполняется асинхронно.

Опросчик агента будет пытаться подключиться ко всем адресам, возвращенным DNS-запросом. Это гарантирует, что если один IP-адрес недоступен, опросчик попробует следующий доступный адрес, повышая вероятность успешного подключения. Это улучшение применяется как к серверу Zabbix, так и к прокси.

Максимальная параллельность асинхронных проверок составляет 1000 (определяется параметром MaxConcurrentChecksPerPoller).

Количество асинхронных опросчиков агента определяется параметром StartAgentPollers.

Запрос сервера

Для определения длины заголовка и данных см. детали протокола.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	"passive checks"
data		*array of object*	yes	Элемент данных пассивной проверки.
	key	*string*	yes	Ключ элемента данных с раскрытыми макросами.
	timeout	*number*	yes	Таймаут связи.

Ответ агента

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	Номер версии агента.
variant		*number*	yes	Вариант агента (*1* - Zabbix agent, *2* - Zabbix agent 2).
data		*array of object*	yes	Содержит результат проверки.
	value	*string*	no	Значение элемента данных, если проверка выполнена успешно.
	error	*string*	no	Сообщение об ошибке, если проверка выполнена неуспешно.

Например, для поддерживаемых элементов данных:

1. Сервер открывает TCP-соединение
2. Сервер отправляет **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}**
3. Агент читает запрос и отвечает **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”value”:1}]}**
4. Сервер обрабатывает данные, чтобы получить значение, в нашем случае — 1
5. TCP-соединение закрывается

Для неподдерживаемых элементов данных:

1. Сервер открывает TCP-соединение
2. Сервер отправляет **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”vfs.fs.size[/nono]”,”timeout”:3**
3. Агент читает запрос и отвечает **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”error”:”Unsupported item key.”}]}**
4. Сервер обрабатывает данные, изменяет состояние элемента данных на неподдерживаемое с указанным сообщением об ошибке
5. TCP-соединение закрывается

Переключение на старый протокол

Чтобы обеспечить работу сервера или прокси Zabbix с агентами версий до 7.2, которые используют протокол в открытом виде, реализован переход на старый протокол.

Пассивные проверки выполняются с использованием JSON-протокола (7.0 и новее) после перезапуска или при изменении конфигурации интерфейса.
Если в ответ не получен допустимый JSON (агент отправил ”ZBX_NOTSUPPORTED”), Zabbix кэширует интерфейс как использующий старый протокол и **повторяет** проверку, отправляя только ключ элемента данных.

Обратите внимание, что каждый час сервер/прокси Zabbix снова будет пытаться работать с новым протоколом со всеми интерфейсами, при необходимости возвращаясь к старому протоколу.

Активные проверки

Активные проверки требуют более сложной обработки.
Сначала агент должен получить от сервера/прокси список элементов данных и/или удаленных команд для независимой обработки.

Серверы/прокси, от которых необходимо получать активные проверки, указаны в параметре `ServerActive` файла конфигурации агента.
Периодичность запроса этих проверок задается параметром `RefreshActiveChecks` в том же файле конфигурации.

Однако если обновить активные проверки не удается, повторная попытка выполняется через 60 секунд, заданных в коде.

> **Note:**
> Начиная с Zabbix 6.4 агент (в активном режиме) больше не получает от сервера/прокси полную копию конфигурации раз в две минуты (по умолчанию). Вместо этого, чтобы снизить сетевой трафик и расход ресурсов, каждые 5 секунд (по умолчанию) выполняется инкрементальная синхронизация конфигурации, при которой сервер/прокси предоставляет полную копию конфигурации **только** если агент еще не получил ее либо если что-то изменилось в конфигурации узла сети, глобальных макросах или глобальных регулярных выражениях.

Агент затем периодически отправляет новые значения на сервер(ы). Если агент получил какие-либо удаленные команды для выполнения, результат выполнения также будет отправлен. Обратите внимание, что выполнение удаленных команд на активном агенте поддерживается начиная с Zabbix agent 7.0.

> **Note:**
> Если агент находится за межсетевым экраном, вы можете рассмотреть использование только активных проверок, поскольку в этом случае вам не потребуется изменять настройки межсетевого экрана, чтобы разрешить первоначальные входящие соединения.

Получение списка элементов данных

Запрос агента

Запрос active checks используется для получения активных проверок, которые должны обрабатываться агентом. Этот запрос отправляется агентом при запуске, а затем с интервалами RefreshActiveChecks.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Имя узла сети.
host_metadata	*string*	no	Значение конфигурационного параметра HostMetadata или метрики HostMetadataItem.
interface	*string*	no	Значение конфигурационного параметра HostInterface или метрики HostInterfaceItem.
ip	*string*	no	Первый IP-адрес параметра конфигурации ListenIP, если он задан.
port	*number*	no	Значение параметра конфигурации ListenPort, если он задан и не является портом прослушивания агента по умолчанию.
version	*string*	yes	Номер версии агента.
variant	*number*	yes	Вариант агента (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Идентификатор конфигурации для инкрементальной синхронизации конфигурации.
session	*string*	no	Идентификатор сеанса для инкрементальной синхронизации конфигурации.

Ответ сервера

Ответ active checks отправляется сервером агенту после обработки запроса active checks.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
```

```
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Информация об ошибке в случае сбоя.
data	*array of objects*	no	Элементы данных активных проверок. Не указывается, если конфигурация узла сети не изменилась.
key	*string*	no	Ключ элемента данных с раскрытыми макросами.
itemid	*number*	no	Идентификатор элемента данных.
delay	*string*	no	Интервал обновления элемента данных. Гибкие интервалы/интервалы по расписанию поддерживаются как Zabbix agent, так и Zabbix agent 2 начиная с Zabbix 7.0.
lastlogsize	*number*	no	lastlogsize элемента данных.
mtime	*number*	no	mtime элемента данных.
timeout	*string*	no	Тайм-аут элемента данных.
refresh_unsupported	*number*	no	Интервал обновления неподдерживаемого элемента данных.
regexp	*array of objects*	no	Глобальные регулярные выражения.
name	*string*	no	Имя глобального регулярного выражения.
expression	*string*	no	Глобальное регулярное выражение.
expression_type	*number*	no	Тип глобального регулярного выражения.
exp_delimiter	*string*	no	Разделитель глобального регулярного выражения.
case_sensitive	*number*	no	Параметр чувствительности глобального регулярного выражения к регистру.
commands	*array of objects*	no	Удаленные команды для выполнения. Включаются, если выполнение удаленной команды было инициировано операцией действия или ручным выполнением скрипта. Обратите внимание, что выполнение удаленных команд на активном агенте поддерживается начиная с Zabbix agent 7.0. Более старые активные агенты будут игнорировать любые удаленные команды, включенные в ответ сервера active checks.
command	*string*	no	Удаленная команда.
id	*number*	no	Идентификатор удаленной команды.
wait	*number*	no	Режим выполнения удаленной команды (”0” (nowait) для команд из операций действия; ”1” (wait) для команд из ручного выполнения скрипта).
timeout	*number*	no	Тайм-аут выполнения удаленной команды в конфигурации сервера/прокси.

Field	Type	Mandatory	Value
config_revision	*number*	no	Идентификатор конфигурации для инкрементальной синхронизации конфигурации. Не указывается, если конфигурация узла сети не изменилась. Увеличивается, если конфигурация узла сети изменена.

Сервер должен отвечать success.

Например:

1. Агент открывает TCP-соединение
2. Агент запрашивает список проверок
3. Сервер отвечает списком элементов данных и удаленных команд для выполнения
4. Агент разбирает ответ
5. TCP-соединение закрывается
6. Агент начинает периодический сбор данных и выполняет удаленные команды (поддерживается начиная с Zabbix agent 7.0)

Attention:
Обратите внимание, что конфиденциальные данные конфигурации могут стать доступными сторонам, имеющим доступ к порту trapper сервера Zabbix, при использовании активной проверки. Это возможно, поскольку любой может выдать себя за активный агент и запросить данные конфигурации элемента данных; аутентификация не выполняется, если не используются параметры шифрования.

Отправка собранных данных

Агент отправляет

Запрос данных агента содержит собранные значения элементов данных и значения выполненных удаленных команд (если они есть).

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

Поле	Тип	Обязательное	Значение
request	*string*	yes	`agent data`

Поле	Тип	Обязательно	Значение
data	*array of objects*	yes	Значения элементов данных.
id	*number*	yes	Идентификатор значения (инкрементный счетчик, используемый для проверки дублирующихся значений в случае проблем с сетью).
itemid	*number*	yes	Идентификатор элемента данных.
value	*string*	no	Значение элемента данных.
lastlogsize	*number*	no	lastlogsize элемента данных.
mtime	*number*	no	mtime элемента данных.
state	*number*	no	Состояние элемента данных.
source	*string*	no	Источник события журнала событий значения.
eventid	*number*	no	eventid события журнала событий значения.
severity	*number*	no	severity события журнала событий значения.
timestamp	*number*	no	timestamp события журнала событий значения.
clock	*number*	yes	Метка времени значения (секунды с начала эпохи).
ns	*number*	yes	Наносекунды метки времени значения.
commands	*array of objects*	no	Результат выполнения удаленных команд. Обратите внимание, что выполнение удаленных команд на активном агенте поддерживается начиная с Zabbix agent 7.0. Более старые активные агенты будут игнорировать любые удаленные команды, включенные в ответ сервера на активные проверки.
id	*number*	no	Идентификатор удаленной команды.
value	*string*	no	Результат выполнения удаленной команды, если выполнение было успешным.
error	*string*	no	Сообщение об ошибке выполнения удаленной команды, если выполнение завершилось неудачей.
session	*string*	yes	Уникальный идентификатор сеанса, генерируемый каждый раз при запуске агента (32 шестнадцатеричных символа).
host	*string*	yes	Имя узла сети.
version	*string*	yes	Номер версии агента.
variant	*number*	yes	Вариант агента (*1* - Zabbix agent, *2* - Zabbix agent 2).

Виртуальный ID назначается каждому значению. ID значения — это простой возрастающий счетчик, уникальный в рамках одного сеанса данных (идентифицируемого по токену сеанса). Этот ID используется для отбрасывания дублирующихся значений, которые могут быть отправлены в условиях плохого качества связи.

Ответ сервера

Ответ на данные агента отправляется сервером обратно агенту после обработки запроса данных агента.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Поле	Тип	Обязательно	Значение
response	*string*	yes	`success` | `failed`
info	*string*	yes	Результаты обработки элементов данных.

Attention:
Если отправка некоторых значений на сервере завершается неудачей (например, потому что узел сети или элемент данных был отключен или удален), агент не будет повторять отправку этих значений.

Например:

1. Агент открывает TCP-соединение
2. Агент отправляет список значений
3. Сервер обрабатывает данные и отправляет статус обратно
4. TCP-соединение закрывается

Attention:
Сообщение об ошибке будет обрезано до 2048 символов на стороне сервера.

Сообщение проверки доступности

Агент отправляет

Сообщение проверки доступности отправляется активным агентом на сервер Zabbix или прокси каждые `HeartbeatFrequency` секунд (параметр настраивается в конфигурационном файле агента Zabbix или агента Zabbix 2).

Оно используется для мониторинга доступности активных проверок.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Поле	Тип	Обязательный	Значение
request	*string*	да	`active check heartbeat`
host	*string*	да	Имя узла сети.
heartbeat_freq	*number*	да	Частота проверки доступности агентом (параметр конфигурации HeartbeatFrequency).
version	*string*	да	Номер версии агента.
variant	*number*	да	Вариант агента (*1* - агент Zabbix, *2* - агент Zabbix 2).

Ответ с перенаправлением

Если узел сети был переназначен, сервер может указать агенту перенаправить сообщения проверки доступности (и последующие активные проверки) на другой экземпляр прокси или сервера.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.1:10055"
    }
  }
```

Поле		Тип	Обязательный	Значение
response		*string*	да	`success` | `failed`
redirect		*object*	да	Инструкции по перенаправлению.
	revision	*number*	да	Идентификатор редакции конфигурации.
	address	*string*	да	Адрес целевого сервера или прокси.

Старый протокол XML

Note:
Zabbix принимает до 16 МБ данных в формате XML, закодированных в Base64, однако размер одного декодированного значения не должен превышать 64 КБ; в противном случае при декодировании оно будет усечено до 64 КБ.

Запуск агента от имени root

Начиная с Zabbix **5.0.0**, файл службы systemd для агента Zabbix в официальных пакетах явно содержит директивы `User` и `Group`. Обе установлены в `zabbix`.

Теперь невозможно настроить, от имени какого пользователя запускается агент Zabbix, через файл `zabbix_agentd.conf`, поскольку агент будет игнорировать эту настройку и запускаться от имени пользователя, указанного в файле службы systemd. Чтобы запускать агент Zabbix от имени root, необходимо внести изменения, описанные ниже.

Агент Zabbix

Чтобы переопределить пользователя и группу по умолчанию для агента Zabbix, выполните:

```
systemctl edit zabbix-agent
```

Затем добавьте следующее содержимое:

```
[Service]
User=root
Group=root
```

Перезагрузите службы и перезапустите службу zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Для **агента Zabbix** это повторно включает возможность настройки пользователя в файле конфигурации агента. Теперь необходимо задать параметры конфигурации `User=root` и `AllowRoot=1` в файле конфигурации.

Zabbix agent 2

Чтобы переопределить пользователя и группу по умолчанию для Zabbix agent 2, выполните:

```
systemctl edit zabbix-agent2
```

Затем добавьте следующий содержимое:

```
[Service]
User=root
Group=root
```

Перезагрузите демоны и перезапустите службу zabbix-agent2:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Для **Zabbix agent2** это полностью определяет пользователя, от имени которого он запускается. Дополнительные изменения не требуются.

Параметры конфигурации: Zabbix агент

Обзор

Параметры конфигурации позволяют настраивать поведение агента Zabbix.

Укажите значения параметров в файле конфигурации агента Zabbix (`zabbix_agentd.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр является обязательным.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его имя)
- Подробные сведения о параметрах

Параметр	Описание
Alias	Задать псевдоним для ключа элемента данных.
AllowKey	Разрешить выполнение ключей элементов данных, соответствующих шаблону.
AllowKeyRegexp	Разрешить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
AllowRoot	Разрешить агенту работать от имени `root`.
BufferSend	Задать частоту отправки значений из буфера на сервер/прокси Zabbix.
BufferSize	Задать максимальное количество значений в буфере памяти.
DebugLevel	Задать уровень отладки.
DenyKey	Запретить выполнение ключей элементов данных, соответствующих шаблону.
DenyKeyRegexp	Запретить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
EnableRemoteCommands	Разрешить удаленные команды от сервера Zabbix. Этот параметр устарел; вместо него используйте `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.
HeartbeatFrequency	Задать частоту сообщений проверки доступности в секундах.
HostInterface	Указать интерфейс узла сети.
HostInterfaceItem	Указать элемент данных для получения интерфейса узла сети.

Параметр	Описание
HostMetadata	Указать метаданные узла сети.
HostMetadataItem	Указать элемент данных агента Zabbix для получения метаданных узла сети.
Hostname	Указать имя узла сети.
HostnameItem	Указать элемент данных агента Zabbix для получения имени узла сети.
Include	Указать каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
ListenBacklog	Задать максимальное количество ожидающих подключений в очереди TCP.
ListenIP	Указать IP-адрес, на котором агент ожидает подключения от сервера/прокси.
ListenPort	Указать порт, на котором агент ожидает подключения.
LoadModule	Указать модуль, который необходимо загрузить при запуске агента.
LoadModulePath	Указать полный путь к расположению модулей агента.
LogFile	Указать файл журнала агента.
LogFileSize	Задать максимальный размер файла журнала.
LogRemoteCommands	Включить запись выполненных команд оболочки в журнал как предупреждений.
LogType	Указать тип вывода журнала.
MaxLinesPerSecond	Задать максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок `log` или `logrt`.
PidFile	Указать файл PID.
RefreshActiveChecks	Указать частоту обновления списка активных проверок.
Server	Указать IP-адреса или DNS-имена серверов/прокси Zabbix.
ServerActive	Указать адрес сервера/прокси Zabbix или конфигурацию кластера, откуда необходимо получать активные проверки.
SourceIP	Указать исходный IP-адрес.
StartAgents	Задать количество предварительно разветвленных экземпляров `zabbix_agentd` для обработки пассивных проверок.
Timeout	Указать, как долго (в секундах) ожидать установления соединения и обмена данными с прокси или сервером Zabbix.
TLSAccept	Указать требуемый уровень шифрования для входящих подключений.
TLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации для проверки сертификатов узлов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCertFile	Указать полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCipherAll	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Указать строку шифров для OpenSSL 1.1.1 или более поздней версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Указать строку шифров для OpenSSL 1.1.1 или более поздней версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Указать строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Указать строку шифров для OpenSSL 1.1.1 или более поздней версии в TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Указать требуемый уровень шифрования для подключений к серверу или прокси Zabbix.
TLSCRLFile	Указать полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного обмена данными между компонентами Zabbix.
TLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованного обмена данными между компонентами Zabbix.
TLSPSKFile	Указать полный путь к файлу, содержащему предварительно установленный ключ агента, используемый для зашифрованного обмена данными с сервером или прокси Zabbix.
TLSPSKIdentity	Указать строку идентификатора предварительно установленного ключа, используемую для зашифрованного обмена данными с сервером или прокси Zabbix.

Параметр	Описание
TLSServerCertIssuer	Указать разрешенного издателя сертификата сервера (прокси).
TLSServerCertSubject	Указать разрешенную тему сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу всех символов в аргументах пользовательских параметров.
User	Понизить привилегии до определенного существующего пользователя в системе.
UserParameter	Указать пользовательский параметр для мониторинга.
UserParameterDir	Указать путь поиска по умолчанию для команд UserParameter.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для демона, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

Псевдоним

Задайте псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Можно указать несколько параметров `Alias`. Несколько параметров с одинаковым ключом `Alias` не допускаются.
 Разные ключи `Alias` могут ссылаться на один и тот же ключ элемента данных.
 Псевдонимы можно использовать в параметре `HostMetadataItem`, но не в параметре `HostnameItem`.

Пример 1: получение идентификатора пользователя `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Теперь для получения данных можно использовать сокращенный ключ `zabbix.userid`.

Пример 2: получение загрузки ЦП с параметрами по умолчанию и пользовательскими параметрами.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Это позволяет использовать ключ `cpu.util` для получения процента загрузки ЦП с параметрами по умолчанию, а также `cpu.util[all, idle, avg15]` для получения определенных данных о загрузке ЦП.

Пример 3: запуск нескольких правил обнаружения низкоуровневых объектов, обрабатывающих одни и те же элементы данных обнаружения.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Теперь можно настроить несколько правил обнаружения с использованием `vfs.fs.discovery` и разными параметрами для каждого правила, например `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` и т. д.

AllowKey

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными знаками, которое поддерживает символ * для сопоставления с любым количеством любых символов.
 Можно определить несколько правил сопоставления ключей в сочетании с `DenyKey`, `AllowKeyRegexp` и `DenyKeyRegexp`. Параметры обрабатываются по одному в порядке их следования. Порядок определяет, какое из совпадающих правил будет иметь приоритет. См. также: Ограничение проверок агента.

AllowKeyRegexp

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с `DenyKeyRegexp`, `AllowKey` и `DenyKey`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

Пример:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Разрешает агенту запускаться от имени `root`. Если параметр отключен и агент запущен пользователем `root`, агент попытается переключиться на пользователя `zabbix`. Не имеет эффекта, если агент запущен обычным пользователем.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

BufferSend

Установите частоту отправки значений из буфера на сервер/прокси Zabbix (в секундах).

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Установите максимальное количество значений в буфере памяти.
 Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнится.

По умолчанию: 100
 Диапазон: 2-65535

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

DenyKey

Запрет выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными символами и поддерживает символ * для соответствия любому количеству любых символов.
 Можно определить несколько правил сопоставления ключей в сочетании с `AllowKey`, `AllowKeyRegexp` и `DenyKeyRegexp`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

По умолчанию: `DenyKey=system.run[*]`

DenyKeyRegexp

Запретить выполнение тех элементов данных, ключи которых соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с `AllowKeyRegexp`, `AllowKey` и `DenyKey`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет приоритет правил, соответствующих условию.
 Все элементы данных `system.run[]` по умолчанию запрещены (`DenyKey=system.run[*]`), если не указано другое правило `system.run[*]`.

См. также: Ограничение проверок агентом.

EnableRemoteCommands

Разрешает удаленные команды от сервера Zabbix.
 Этот параметр **устарел**; вместо него используйте `AllowKey=system.run[*]` `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` или `DenyKeyRegexp=^system\.run\[.*\]$`.
 Это внутренний псевдоним параметров `AllowKey`, `AllowKeyRegexp`, `DenyKey` и `DenyKeyRegexp` в зависимости от значения: - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.run\[.*\]$`)

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

HeartbeatFrequency

Установите частоту сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 - сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Укажите интерфейс узла сети (IP-адрес или DNS-имя), используемый во время авторегистрации узла сети. Это значение будет использовано для заполнения интерфейса в вновь созданном узле сети и позволяет явно настроить либо IP-, либо DNS-адрес. Подробнее см. Использование DNS в качестве интерфейса по умолчанию.

Если значение не задано, оно будет получено из `HostInterfaceItem`.

Агент выдаст ошибку и не запустится, если значение превышает лимит в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Укажите элемент данных для определения интерфейса узла сети (IP-адреса или DNS-имени) во время автоматической регистрации узла сети. Это значение используется только в том случае, если параметр `HostInterface` не определен. Дополнительные сведения см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса на автоматическую регистрацию агент запишет предупреждение в журнал, если значение, возвращенное указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

HostMetadata

Укажите метаданные, чтобы идентифицировать или отличить узел сети во время автоматической регистрации узла сети (активный агент). HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если параметр не задан, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или не является строкой в кодировке UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, являющиеся допустимыми строками UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечён на первом переносе строки.

Диапазон: 0–2034 байта

HostMetadataItem

Укажите элемент данных агента Zabbix для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не определен.

Значение HostMetadataItem извлекается при каждой попытке автоматической регистрации и используется только в процессе автоматической регистрации узла сети (активный агент). HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Поддерживаются пользовательские параметры и псевдонимы. Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

Во время запроса на автоматическую регистрацию агент запишет предупреждение в журнал, если значение, возвращенное указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращенное элементом данных, должно быть строкой UTF-8; в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, являющиеся допустимыми строками UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечен на первом переносе строки.

Имя узла сети

Укажите уникальные имена узлов сети с учетом регистра. Список, разделенный запятыми.
 Требуется для активных проверок и должен совпадать с именами узлов сети, настроенными на сервере.
 Значение берется из HostnameItem, если оно не задано.
Допустимые символы: буквенно-цифровые, '.', ' ', '_' и '-'. Максимальная длина: 128 символов на каждое имя узла сети, 2048 символов для всей строки.

По умолчанию: задается HostnameItem

HostnameItem

Укажите элемент данных агента Zabbix для получения имени узла сети.
 Этот параметр используется только в том случае, если Hostname не определен.
 Пользовательские параметры и псевдонимы не поддерживаются, однако элемент данных system.run[] поддерживается независимо от значений AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

Значение по умолчанию: system.hostname

Include

Укажите каталог или отдельные файлы, которые нужно включить в файл конфигурации.
 Во время установки Zabbix этот параметр устанавливается в /usr/local/etc, если вы не измените его на этапе компиляции.

Общие правила для параметра Include:

- Путь может быть относительным к расположению файла zabbix_agentd.conf.
- В одном файле конфигурации можно указать несколько параметров Include.
- Zabbix обрабатывает параметр Include сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.

- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила для включения каталогов или шаблонов файлов:

- Не следует предполагать какой-либо порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общий параметр более специфичным), так как невозможно предсказать, какое значение будет использовано.
- Обратите внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан `include/my_specific.conf.BAK`), поскольку будут включены оба файла.

Чтобы включать только нужные файлы из каталога, используйте подстановочный знак (*) для сопоставления с шаблоном.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Установите максимальное число ожидающих соединений в очереди TCP.
 Значение по умолчанию — жестко заданная константа, которая зависит от системы.
 Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть незаметно усечены до 'implementation-specified maximum'.

По умолчанию: SOMAXCONN
 Диапазон: 0 - INT_MAX

ListenIP

Укажите IP-адрес, на котором агент принимает подключения от сервера/прокси. Список значений, разделенных запятыми.

По умолчанию: 0.0.0.0

ListenPort

Укажите порт прослушивания агента для подключений от сервера/прокси.

По умолчанию: 10050
 Диапазон: 1024-32767

LoadModule

Укажите модуль, который нужно загрузить при запуске агента.
 Модули используются для расширения функциональности агента.
 Модуль должен находиться в каталоге, указанном в LoadModulePath, либо путь должен предшествовать имени модуля. Если указанный путь является абсолютным (начинается с /), то LoadModulePath игнорируется.
 Форматы: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

Допускается указывать несколько параметров LoadModule.

LoadModulePath

Укажите полный путь к расположению модулей агента.
 Значение по умолчанию зависит от параметров компиляции.

LogFile

Укажите файл журнала агента.

Обязательно: Да, если LogType установлен в file; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: Если достигается ограничение размера файла журнала и ротация файла по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включить журналирование выполненных команд оболочки как предупреждений.
 Команды будут записываться в журнал только при удаленном выполнении.
 Записи в журнале не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *system* - записывать журнал в syslog; - *console* - записывать журнал в стандартный вывод.

По умолчанию: `file`

MaxLinesPerSecond

Установите максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок `log` или `logrt`.
 Указанное значение будет переопределено параметром `maxlines`, заданным в ключе элемента данных `log` или `logrt`.
Примечание: Zabbix будет обрабатывать в 10 раз больше новых строк, чем задано в `MaxLinesPerSecond`, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

PidFile

Укажите PID-файл.

По умолчанию: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Укажите частоту обновления списка активных проверок в секундах.
 Обратите внимание, что после неудачной попытки обновить активные проверки следующая попытка обновления будет выполнена через 60 секунд.

По умолчанию: 5
 Диапазон: 1-86400

Сервер

Укажите IP-адреса, при необходимости в нотации CIDR, или DNS-имена серверов/прокси Zabbix. Список, разделенный запятыми.
 Входящие подключения будут приниматься только от узлов сети, перечисленных здесь. Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` обрабатываются одинаково, и `::/0` позволит любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес. Обратите внимание, что "IPv4-compatible IPv6 addresses" (`0000::/96 prefix`) поддерживаются, но устарели согласно RFC4291. Допускаются пробелы.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательно: да, если StartAgents явно не задано в 0

ServerActive

Укажите адрес сервера/прокси Zabbix или конфигурацию кластера, откуда получать активные проверки.
 Адрес сервера/прокси — это IP-адрес или DNS-имя, а также необязательный порт, отделенный двоеточием.
Конфигурация кластера — это один или несколько адресов участников группы серверов или прокси, разделенных точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделяя их запятыми. Если не используются группы прокси, для каждого сервера/кластера Zabbix не следует указывать более одного прокси Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделенных запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Пробелы допускаются.
Если порт не указан, используется порт по умолчанию.
IPv6-адреса должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для IPv6-адресов необязательны.
Если этот параметр не указан, активные проверки отключены.

Пример для прокси Zabbix:

```
ServerActive=127.0.0.1:10051
```

Пример для группы прокси Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Пример для нескольких серверов:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Пример для высокой доступности:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Пример для высокой доступности с двумя кластерами и одним сервером:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Укажите исходящий IP-адрес для:

- исходящих подключений к серверу Zabbix или прокси Zabbix;
- установления соединений при выполнении некоторых элементов данных (`web.page.get`, `net.tcp.port` и т. д.).

StartAgents

Установите количество предварительно созданных экземпляров `zabbix_agentd` для обработки пассивных проверок.
 Если установлено значение 0, пассивные проверки отключаются, и агент не будет прослушивать какой-либо TCP-порт.

По умолчанию: 10
 Диапазон: 0-100

Таймаут

Укажите, сколько времени ожидать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций связи, включая:

- ожидание ответа от сервера Zabbix;
- отправку запросов на сервер Zabbix, включая запросы на настройку элементов данных и запросы данных элементов данных в активных проверках;
- получение данных журнала через logfile;
- отправку heartbeat-сообщений;
- максимальную длительность проверок `vfs.*`;
- использование модулями агента Zabbix;
- выполнение роли резервного значения в сценариях, когда сервер или прокси старше версии 7.0 отправляет проверки без таймаутов.

Этот таймаут **не** будет использоваться для тех проверок агента, для которых в веб-интерфейсе можно настроить значение таймаута (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Укажите уровень шифрования, требуемый для входящих подключений.
 Используется для пассивных проверок.
 Можно указать несколько значений, разделяя их запятыми: - *unencrypted* - принимать подключения без шифрования (по умолчанию); - *psk* - принимать подключения с TLS и предварительно общим ключом (PSK); - *cert* - принимать подключения с TLS и сертификатом.

Обязательно: да, если определены параметры TLS certificate или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCAFile

Укажите полное имя пути к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата узла-пира, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованной связи с компонентами Zabbix.

TLSCipherAll

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Укажите строку шифров для OpenSSL 1.1.1 или более новой версии в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Укажите строку шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

TLSCipherPSK

Укажите строку приоритетов GnuTLS или строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример для GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Пример для OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Укажите строку шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Укажите уровень шифрования, требуемый для подключений агента к серверу Zabbix или прокси.
 Используется для активных проверок.
 Можно указать только одно значение: - *unencrypted* - подключение без шифрования (по умолчанию); - *psk* - подключение с использованием TLS и предварительно общего ключа (PSK); - *cert* - подключение с использованием TLS и сертификата.

Обязательно: да, если заданы параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCRLFile

Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованной связи между компонентами Zabbix.

TLSPSKFile

Укажите полный путь к файлу, содержащему предварительно общий ключ агента, используемый для зашифрованной связи с сервером Zabbix или прокси.

TLSPSKIdentity

Укажите строку идентификатора предварительно разделяемого ключа, используемую для зашифрованной связи с сервером или прокси Zabbix.

TLSServerCertIssuer

Укажите разрешенного издателя сертификата сервера (прокси).

TLSServerCertSubject

Укажите допустимый subject сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передавать в аргументах пользовательских параметров все символы. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не допускаются.

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

Пользователь

Снижает привилегии до конкретного существующего пользователя в системе.
Имеет эффект только при запуске от `root`, если `AllowRoot` отключен.

По умолчанию: `zabbix`

UserParameter

Укажите пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: `UserParameter=<key>,<shell command>`
Обратите внимание, что команда оболочки не должна возвращать пустую строку или только EOL. Команды оболочки могут использовать относительные пути, если указан параметр `UserParameterDir`.

Пример:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Укажите путь поиска по умолчанию для команд `UserParameter`. Если этот параметр используется, агент изменит свой рабочий каталог на указанный здесь перед выполнением команды. Таким образом, команды `UserParameter` могут использовать относительный префикс `./` вместо полного пути.
Допускается только одна запись.

Пример:

```
UserParameterDir=/opt/myscripts
```

См. также

1. Различия в конфигурации агента Zabbix для активных и пассивных проверок, начиная с версии 2.0.0

Параметры конфигурации: Zabbix агент (Windows)

Обзор

Параметры конфигурации позволяют настраивать поведение агента Zabbix в Windows.

Укажите значения параметров в файле конфигурации агента Zabbix (`zabbix_agentd.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его название).
- Подробные сведения о параметрах.

Параметр	Описание
Alias	Задать псевдоним для ключа элемента данных.
AllowKey	Разрешить выполнение ключей элементов данных, соответствующих шаблону.
AllowKeyRegexp	Разрешить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
BufferSend	Задать частоту отправки значений из буфера на сервер/прокси Zabbix.
BufferSize	Задать максимальное количество значений в буфере памяти.
DebugLevel	Задать уровень отладки.
DenyKey	Запретить выполнение ключей элементов данных, соответствующих шаблону.
DenyKeyRegexp	Запретить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
EnableRemoteCommands	Разрешить удаленные команды от сервера Zabbix. Этот параметр устарел; вместо него используйте `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.
HeartbeatFrequency	Задать частоту отправки сообщений проверки доступности в секундах.
HostInterface	Указать интерфейс узла сети.
HostInterfaceItem	Указать элемент данных для получения интерфейса узла сети.
HostMetadata	Указать метаданные узла сети.
HostMetadataItem	Указать элемент данных агента Zabbix для получения метаданных узла сети.
Hostname	Указать имя узла сети.
HostnameItem	Указать элемент данных агента Zabbix для получения имени узла сети.
Include	Указать каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
ListenBacklog	Задать максимальное количество ожидающих подключений в очереди TCP.
ListenIP	Указать IP-адрес прослушивания агента для подключений от сервера/прокси.

Параметр	Описание
ListenPort	Указать порт прослушивания агента.
LogFile	Указать файл журнала агента.
LogFileSize	Задать максимальный размер файла журнала.
LogRemoteCommands	Включить запись выполненных команд оболочки в журнал как предупреждений.
LogType	Указать тип вывода журнала.
MaxLinesPerSecond	Задать максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок `log`, `logrt` или `eventlog`.
PerfCounter	Определить новый параметр <parameter_name>, представляющий собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период <period> (в секундах).
PerfCounterEn	Определить новый параметр <parameter_name>, представляющий собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период <period> (в секундах). В отличие от PerfCounter, пути к счетчикам производительности должны быть указаны на английском языке.
RefreshActiveChecks	Указать частоту обновления списка активных проверок.
Server	Указать IP-адреса или DNS-имена серверов/прокси Zabbix.
ServerActive	Указать адрес сервера/прокси Zabbix или конфигурацию кластера, откуда необходимо получать активные проверки.
SourceIP	Указать исходный IP-адрес.
StartAgents	Задать количество предварительно запущенных экземпляров `zabbix_agentd` для обработки пассивных проверок.
Timeout	Указать, как долго (в секундах) ожидать установления соединения и обмена данными с сервером или прокси Zabbix.
TLSAccept	Указать требуемый уровень шифрования для входящих подключений.
TLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации для проверки сертификата узла, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCertFile	Указать полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSConnect	Указать требуемый уровень шифрования для подключений к серверу или прокси Zabbix.
TLSCRLFile	Указать полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного обмена данными между компонентами Zabbix.
TLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованного обмена данными между компонентами Zabbix.
TLSPSKFile	Указать полный путь к файлу, содержащему предварительно установленный ключ агента, используемый для зашифрованного обмена данными с сервером или прокси Zabbix.
TLSPSKIdentity	Указать строку идентификатора предварительно установленного ключа, используемую для зашифрованного обмена данными с сервером или прокси Zabbix.
TLSServerCertIssuer	Указать разрешенного издателя сертификата сервера (прокси).
TLSServerCertSubject	Указать разрешенную тему сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу всех символов в аргументах пользовательских параметров.
UserParameter	Указать пользовательский параметр для мониторинга.
UserParameterDir	Указать путь поиска по умолчанию для команд UserParameter.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для демона, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

Псевдоним

Задайте псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Можно указать несколько параметров `Alias`. Несколько параметров с одинаковым ключом `Alias` не допускаются.
 Разные ключи `Alias` могут ссылаться на один и тот же ключ элемента данных.
 Псевдонимы можно использовать в параметре `HostMetadataItem`, но не в параметрах `HostnameItem` или `PerfCounter`.

Пример 1: получение процента использования файла подкачки на сервере.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Теперь для получения данных можно использовать сокращенный ключ pg_usage.

Пример 2: получение загрузки ЦП со стандартными и пользовательскими параметрами.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Это позволяет использовать ключ cpu.load для получения загрузки ЦП со стандартными параметрами, а также cpu.load[percpu,avg15] для получения определенных данных о загрузке ЦП.

Пример 3: запуск нескольких правил обнаружения низкого уровня, обрабатывающих одни и те же элементы данных обнаружения.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Теперь можно настроить несколько правил обнаружения с использованием vfs.fs.discovery и разными параметрами для каждого правила, например vfs.fs.discovery[foo], vfs.fs.discovery[bar] и т. д.

AllowKey

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными знаками, которое поддерживает символ * для сопоставления с любым количеством любых символов.
 Несколько правил сопоставления ключей могут быть определены в сочетании с DenyKey, AllowKeyRegexp и DenyKeyRegexp. Параметры обрабатываются по одному в порядке их следования. Порядок определяет, какое из совпадающих правил будет иметь приоритет. См. также: Ограничение проверок агента.

AllowKeyRegexp

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с DenyKeyRegexp, AllowKey и DenyKey. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

Пример:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Установите частоту отправки значений из буфера на сервер/прокси Zabbix (в секундах).

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Установите максимальное количество значений в буфере памяти.
 Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнится.

По умолчанию: 100
 Диапазон: 2-65535

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

DenyKey

Запрещает выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными символами и поддерживает символ *, соответствующий любому количеству любых символов.
 Можно определить несколько правил сопоставления ключей в сочетании с AllowKey, AllowKeyRegexp и DenyKeyRegexp. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

По умолчанию: DenyKey=system.run[*]

DenyKeyRegexp

Запрет выполнения тех элементов данных, ключи которых соответствуют шаблону. Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с AllowKeyRegexp, AllowKey и DenyKey. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет.
 Все элементы данных system.run[] по умолчанию запрещены (DenyKey=system.run[*]), если не указано другое правило system.run[*].

См. также: Ограничение проверок агента.

EnableRemoteCommands

Разрешает удаленные команды от сервера Zabbix.
 Этот параметр **устарел**; вместо него используйте AllowKey=system.run[*] AllowKeyRegexp=^system\.run\[.*\]$, DenyKey=system.run[*] или DenyKeyRegexp=^system\.run\[.*\]$.
 Это внутренний псевдоним параметров AllowKey, AllowKeyRegexp, DenyKey и DenyKeyRegexp в зависимости от значения: - 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.run\[.*\]$)

По умолчанию: 0
 Значения: 0 - запрещено, 1 - разрешено

HeartbeatFrequency

Установите частоту отправки сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 - сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Укажите интерфейс узла сети (IP-адрес или DNS-имя), используемый во время авторегистрации узла сети. Это значение будет использовано для заполнения интерфейса в вновь созданном узле сети и позволяет явно настроить либо IP-, либо DNS-адрес. Подробнее см. Использование DNS в качестве интерфейса по умолчанию.

Если не задано, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает лимит в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Укажите элемент данных для определения интерфейса узла сети (IP-адреса или DNS-имени) во время автоматической регистрации узла сети. Это значение используется только в том случае, если параметр HostInterface не определен. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса на автоматическую регистрацию агент зарегистрирует предупреждение, если значение, возвращенное указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

HostMetadata

Укажите метаданные, используемые для идентификации или различения узла сети во время автоматической регистрации узла сети (активный агент). HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если параметр не задан, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или не является строкой в кодировке UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, являющиеся допустимыми строками UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечён на первом переносе строки.

Диапазон: 0–2034 байта

HostMetadataItem

Укажите элемент данных агента Zabbix для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не определен.

Значение HostMetadataItem извлекается при каждой попытке автоматической регистрации и используется только в процессе автоматической регистрации узла сети (активный агент). HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Поддерживаются пользовательские параметры, счетчики производительности и псевдонимы. Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

Во время запроса на автоматическую регистрацию агент запишет предупреждение в журнал, если значение, возвращенное указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращенное элементом данных, должно быть строкой в кодировке UTF-8; в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми строками UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечен на первом переносе строки.

Имя узла сети

Укажите уникальные имена узлов сети с учетом регистра. Список, разделенный запятыми.
 Требуется для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Значение берется из `HostnameItem`, если не задано.
Допустимые символы: буквенно-цифровые, '.', ' ', '_' и '-'. Максимальная длина: 128 символов на каждое имя узла сети, 2048 символов для всей строки.

По умолчанию: задается через HostnameItem

HostnameItem

Укажите элемент данных агента Zabbix для получения имени узла сети.
 Этот параметр используется только в том случае, если `Hostname` не определен.
 Пользовательские параметры, счетчики производительности и псевдонимы не поддерживаются, однако элемент данных `system.run[]` поддерживается независимо от значений `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.
См. также более подробное описание.

По умолчанию: `system.hostname`

Include

Укажите каталог или отдельные файлы, которые необходимо включить в файл конфигурации (по умолчанию он находится в `C:\Program Files\Zabbix Agent`, если агент Zabbix установлен с помощью пакетов установщика Windows MSI; если агент Zabbix установлен из zip-архива, файл находится в папке, указанной во время установки).

Общие правила для параметра `Include`:

- Путь может быть относительным к расположению файла `zabbix_agentd.conf`.
- В одном файле конфигурации можно указать несколько параметров `Include`.
- Zabbix обрабатывает параметр `Include` сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила включения каталогов или шаблонов файлов:

- Не следует рассчитывать на определенный порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общую настройку специальной), поскольку невозможно предсказать, какое значение будет использовано.
- Обращайте внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан файл `include/my_specific.conf.BAK`), поскольку будут включены оба файла.

Чтобы включить из каталога только нужные файлы, используйте подстановочный знак (`*`) для сопоставления с шаблоном.

Пример:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Установите максимальное число ожидающих соединений в очереди TCP.
 Значение по умолчанию — жестко заданная константа, которая зависит от системы.
 Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть незаметно усечены до 'implementation-specified maximum'.

По умолчанию: `SOMAXCONN`
 Диапазон: 0 - INT_MAX

ListenIP

Укажите IP-адрес, на котором агент принимает подключения от сервера/прокси. Список значений, разделенных запятыми.

По умолчанию: `0.0.0.0`

ListenPort

Укажите порт прослушивания агента для подключений от сервера/прокси.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Укажите файл журнала агента.

По умолчанию: c:\zabbix_agentd.log
 Обязательно: Да, если LogType установлен в file; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: Если достигается ограничение размера файла журнала и ротация файла по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включить журналирование выполненных команд оболочки как предупреждений.
 Команды будут записываться в журнал только при удаленном выполнении.
 Записи в журнале не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *system* - записывать журнал в журнал событий Windows; - *console* - записывать журнал в стандартный вывод.

По умолчанию: file

MaxLinesPerSecond

Установите максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок log, logrt или eventlog.
 Указанное значение будет переопределено параметром maxlines, заданным в ключе элемента данных log, logrt или eventlog.
Примечание: Zabbix будет обрабатывать в 10 раз больше новых строк, чем задано в MaxLinesPerSecond, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

PerfCounter

Определите новый параметр <parameter_name>, который будет представлять собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах).
Синтаксис: <parameter_name>,"<perf_counter_path>",<period>

Например, если вы хотите получать среднее количество прерываний процессора в секунду за последнюю минуту, вы можете определить новый параметр "interrupts" следующим образом:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Обратите внимание на двойные кавычки вокруг пути к счетчику производительности. Имя параметра (interrupts) следует использовать как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения будут сниматься каждую секунду.
Вы можете выполнить typeperf -qx, чтобы получить список всех счетчиков производительности, доступных в Windows.

PerfCounterEn

Определите новый параметр <parameter_name>, который представляет собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах). В отличие от PerfCounter, пути perfcounter должны быть на английском языке. Поддерживается только в **Windows Server 2008/Vista** и более поздних версиях.
Синтаксис: <parameter_name>,"<perf_counter_path>",<period>

Например, если вы хотите получать среднее число прерываний процессора в секунду за последнюю минуту, вы можете определить новый параметр "interrupts" следующим образом:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Обратите внимание на двойные кавычки вокруг пути к счетчику производительности. Имя параметра (interrupts) будет использоваться как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения

будут сниматься каждую секунду.
Список английских строк можно найти, просмотрев следующий раздел реестра: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009.

RefreshActiveChecks

Укажите частоту обновления списка активных проверок в секундах.
 Обратите внимание, что после неудачной попытки обновить активные проверки следующая попытка обновления будет выполнена через 60 секунд.

По умолчанию: 5
 Диапазон: 1-86400

Сервер

Укажите IP-адреса, при необходимости в нотации CIDR, или DNS-имена серверов/прокси Zabbix. Список, разделенный запятыми.
 Входящие соединения будут приниматься только от узлов сети, перечисленных здесь. Если поддержка IPv6 включена, то 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 считаются эквивалентными, и ::/0 позволит использовать любой IPv4- или IPv6-адрес. 0.0.0.0/0 можно использовать, чтобы разрешить любой IPv4-адрес. Обратите внимание, что "IPv4-compatible IPv6 addresses" (0000::/96 prefix) поддерживаются, но считаются устаревшими согласно RFC4291. Допускаются пробелы.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательно: да, если StartAgents явно не задано в 0

ServerActive

Укажите адрес сервера/прокси Zabbix или конфигурацию кластера, откуда получать активные проверки.
 Адрес сервера/прокси — это IP-адрес или DNS-имя, а также необязательный порт, отделенный двоеточием.
 Конфигурация кластера — это один или несколько адресов участников группы серверов или прокси, разделенных точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделяя их запятыми. Если не используются группы прокси, для каждого сервера/кластера Zabbix не следует указывать более одного прокси Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделенных запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Пробелы допускаются.
Если порт не указан, используется порт по умолчанию.
IPv6-адреса должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, для IPv6-адресов квадратные скобки необязательны.
Если этот параметр не указан, активные проверки отключены.

Пример для прокси Zabbix:

```
ServerActive=127.0.0.1:10051
```

Пример для группы прокси Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Пример для нескольких серверов:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Пример для высокой доступности:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Пример для высокой доступности с двумя кластерами и одним сервером:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Диапазон: количество активных серверов, указанных в ServerActive, плюс количество предварительно созданных экземпляров для пассивных проверок, заданных в StartAgents, должно быть меньше 64.

SourceIP

Укажите исходящий IP-адрес для:

- исходящих подключений к серверу Zabbix или прокси Zabbix;
- установления соединений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т. д.).

StartAgents

Установите количество предварительно порожденных экземпляров zabbix_agentd для обработки пассивных проверок.
 Если установлено значение 0, пассивные проверки отключаются, и агент не будет прослушивать какой-либо TCP-порт.

По умолчанию: 10
 Диапазон: 0-100 (*)

Таймаут

Укажите, сколько времени ожидать (в секундах) при установлении соединения и обмене данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций связи, включая:

- ожидание ответа от сервера Zabbix;
- отправку запросов на сервер Zabbix, включая запросы на настройку элементов данных и данные элементов данных в активных проверках;
- получение данных журнала через мониторинг logfile или журнала событий Windows;
- отправку сообщений heartbeat;
- максимальную длительность проверок `vfs.*`;
- использование модулями агента Zabbix;
- использование в качестве резервного варианта в сценариях, когда сервер или прокси старше версии 7.0 отправляет проверки без таймаутов.

Этот таймаут **не** будет использоваться для тех проверок агента, для которых в веб-интерфейсе можно настроить таймаут (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Укажите уровень шифрования, требуемый для входящих подключений.
 Используется для пассивных проверок.
 Можно указать несколько значений, разделяя их запятыми: - *unencrypted* - принимать подключения без шифрования (по умолчанию); - *psk* - принимать подключения с TLS и предварительно общим ключом (PSK); - *cert* - принимать подключения с TLS и сертификатом.

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата узла-пира, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованной связи с компонентами Zabbix.

TLSConnect

Укажите уровень шифрования, требуемый для подключений агента к серверу или прокси Zabbix.
 Используется для активных проверок.
 Можно указать только одно значение: - *unencrypted* - подключение без шифрования (по умолчанию); - *psk* - подключение с использованием TLS и предварительно общего ключа (PSK); - *cert* - подключение с использованием TLS и сертификата.

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCRLFile

Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованной связи между компонентами Zabbix.

TLSPSKFile

Укажите полный путь к файлу, содержащему предварительно общий ключ агента, используемый для зашифрованной связи с сервером Zabbix или прокси.

TLSPSKIdentity

Укажите строку идентификатора предварительно разделенного ключа, используемую для зашифрованной связи с сервером Zabbix или прокси.

TLSServerCertIssuer

Укажите разрешенного издателя сертификата сервера (прокси).

TLSServerCertSubject

Укажите допустимый subject сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передавать все символы в аргументах пользовательских параметров. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не допускаются.

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

UserParameter

Укажите пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: `UserParameter=<key>,<shell command>`
Обратите внимание, что команда оболочки не должна возвращать пустую строку или только EOL. Команды оболочки могут использовать относительные пути, если указан параметр `UserParameterDir`.

Пример:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Укажите путь поиска по умолчанию для команд `UserParameter`. При использовании этого параметра агент изменит свой рабочий каталог на указанный здесь перед выполнением команды. Таким образом, команды `UserParameter` могут использовать относительный префикс `./` вместо полного пути. Допускается только одна запись.

Пример:

```
UserParameterDir=C:\opt\myscripts
```

Zabbix агент в Microsoft Windows

Настройка агента

Оба поколения агентов Zabbix работают как служба Windows.
Для агента Zabbix 2 замените `agentd` на `agent2` в приведенных ниже инструкциях.

На узле сети Microsoft Windows можно запустить один экземпляр агента Zabbix или несколько экземпляров агента.
Один экземпляр может использовать:

- файл конфигурации по умолчанию, расположенный в том же каталоге, что и двоичный файл агента;
- файл конфигурации, указанный в командной строке.

В случае нескольких экземпляров каждый экземпляр агента должен иметь собственный файл конфигурации (один из экземпляров может использовать файл конфигурации по умолчанию).

Пример файла конфигурации доступен в архиве исходного кода Zabbix:

- `conf/zabbix_agentd.conf` для агента Zabbix;
- `conf/zabbix_agent2.conf` для agent2 Zabbix.

Если вы хотите установить агент Zabbix/агент 2 для Windows как службу из архива, не указывая файл конфигурации явно, перед установкой агента выполните следующие действия:

- вручную скопируйте `conf/zabbix_agentd.conf` в каталог, где будет установлен `zabbix_agentd.exe`;
- вручную скопируйте `conf/zabbix_agent2.conf` и каталог `conf/zabbix_agent2.d` в каталог, где будет установлен `zabbix_agent2.exe`.

Подробную информацию о настройке агента Zabbix для Windows см. в параметрах файла конфигурации агента Zabbix или файла конфигурации агента 2 Zabbix.

Параметр Hostname

Для выполнения активных проверок на узле сети агенту Zabbix необходимо иметь определенное имя узла сети. Кроме того, значение имени узла сети, заданное на стороне агента, должно в точности совпадать с параметром *Имя узла сети*, настроенным для узла сети в веб-интерфейсе.

Значение имени узла сети на стороне агента можно определить с помощью параметра `Hostname` или `HostnameItem` в файле конфигурации агента - в противном случае используются значения по умолчанию, если ни один из этих параметров не указан.

Значением по умолчанию для параметра `HostnameItem` является значение, возвращаемое ключом агента `system.hostname`. В Windows он возвращает результат функции `gethostname()`, которая запрашивает поставщиков пространств имен,

чтобы определить локальное имя узла сети. Если ни один поставщик пространств имен не отвечает, возвращается имя NetBIOS.

Значением по умолчанию для `Hostname` является значение, возвращаемое параметром `HostnameItem`. Таким образом, если оба параметра не указаны, фактическим именем узла сети будет имя узла сети NetBIOS; агент Zabbix будет использовать имя узла сети NetBIOS для получения списка активных проверок с сервера Zabbix и отправки ему результатов.

Ключ `system.hostname` поддерживает два необязательных параметра - *type* и *transform*.

Type определяет тип имени, которое должен возвращать элемент данных:

- *netbios* (по умолчанию) - возвращает имя узла сети NetBIOS, ограниченное 15 символами и состоящее только из символов в ВЕРХНЕМ РЕГИСТРЕ;
- *host* - с учетом регистра, возвращает полное фактическое имя узла сети Windows (без домена);
- *shorthost* - возвращает часть имени узла сети до первой точки. Если имя не содержит точки, возвращается полная строка.
- *fqdn* - возвращает полное доменное имя (Fully Qualified Domain Name) без завершающей точки.

Transform позволяет указать дополнительное правило преобразования имени узла сети:

- *none* (по умолчанию) - использовать исходный регистр букв;
- *lower* - преобразовать текст в нижний регистр.

Таким образом, чтобы упростить настройку файла `zabbix_agentd.conf` и сделать ее унифицированной, можно использовать различные подходы:

1. Не указывать параметры `Hostname` или `HostnameItem`, и агент Zabbix будет использовать имя узла сети NetBIOS в качестве имени узла сети.
2. Не указывать параметр `Hostname` и определить `HostnameItem` следующим образом:
    - `HostnameItem=system.hostname[host]` - агент Zabbix будет использовать полное фактическое имя узла сети Windows с учетом регистра в качестве имени узла сети.
    - `HostnameItem=system.hostname[shorthost,lower]` - агент Zabbix будет использовать только часть имени узла сети до первой точки, преобразованную в нижний регистр.
    - `HostnameItem=system.hostname[fqdn]` - агент Zabbix будет использовать полное доменное имя (Fully Qualified Domain Name) в качестве имени узла сети.

Имя узла сети также используется как часть имени службы Windows, которое применяется для установки, запуска, остановки и удаления службы Windows. Например, если в файле конфигурации агента Zabbix указано `Hostname=Windows_db_serve` агент будет установлен как служба Windows `Zabbix Agent [Windows_db_server]`. Поэтому, чтобы для каждого экземпляра агента Zabbix использовать отдельное имя службы Windows, каждый экземпляр должен использовать другое имя узла сети.

Установка агента как службы Windows

Перед установкой агента вручную скопируйте `conf/zabbix_agentd.conf` в каталог, где будет установлен zabbix_agentd.exe.

Чтобы установить один экземпляр агента Zabbix с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --install
```

Attention:
На 64-разрядной системе для корректной работы всех проверок, связанных с запуском 64-разрядных процессов, требуется 64-разрядная версия агента Zabbix.

Если вы хотите использовать файл конфигурации, отличный от файла по умолчанию, для установки службы следует использовать следующую команду:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

Необходимо указать полный путь к файлу конфигурации.

Несколько экземпляров агента Zabbix можно установить как службы следующим образом:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

Установленная служба теперь должна отображаться в Панели управления.

Запуск агента

Чтобы запустить службу агента, можно использовать Панель управления или сделать это из командной строки.

Чтобы запустить один экземпляр агента Zabbix с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --start
```

Чтобы запустить один экземпляр агента Zabbix с другим файлом конфигурации:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Чтобы запустить один из нескольких экземпляров агента Zabbix:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Остановка агента

Чтобы остановить службу агента, вы можете использовать Панель управления или сделать это из командной строки.

Чтобы остановить один экземпляр агента Zabbix, запущенный с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --stop
```

Чтобы остановить один экземпляр агента Zabbix, запущенный с другим файлом конфигурации:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Чтобы остановить один из нескольких экземпляров агента Zabbix:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Удаление службы агента Windows

Чтобы удалить один экземпляр агента Zabbix с использованием файла конфигурации по умолчанию:

```
zabbix_agentd.exe --uninstall
```

Чтобы удалить один экземпляр агента Zabbix с использованием файла конфигурации, отличного от файла по умолчанию:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Чтобы удалить несколько экземпляров агента Zabbix из служб Windows:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

Ограничения

Агент Zabbix для Windows не поддерживает нестандартные конфигурации Windows, в которых процессоры распределены неравномерно между узлами NUMA. Если логические процессоры распределены неравномерно, то метрики производительности CPU могут быть недоступны для некоторых процессоров. Например, если имеется 72 логических процессора и 2 узла NUMA, на каждом узле должно быть по 36 процессоров.

Агент Zabbix в IBM AIX

Обзор

Предварительно скомпилированные двоичные файлы агента Zabbix для IBM AIX доступны для скачивания в виде архива `.tar.gz`. На этой странице описывается установка агента из архива, его запуск вручную или с помощью System Resource Controller (SRC), запуск при загрузке системы и удаление из SRC.

Установка агента

Команда `tar` в AIX не поддерживает параметр `-z`, поэтому распакуйте архив с помощью `gunzip`. Используйте параметр `-o` команды `tar`, чтобы извлеченные файлы принадлежали пользователю, который их извлекает, а не числовым идентификаторам пользователя и группы, сохраненным в архиве. Например, чтобы разместить агент в `/opt/zabbix` (выполните от имени root; каталог `/opt/zabbix` еще не должен существовать):

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

В архиве AIX двоичный файл агента находится в `usr/sbin/`, а файл конфигурации — в `etc/zabbix/`. Перед запуском агента при необходимости измените файл `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` (как минимум параметры `Server`, `ServerActive` и `Hostname`).

Если агент запущен от имени root, он переключается на пользователя zabbix (или пользователя, указанного параметром User), если в файле конфигурации не задано AllowRoot=1. Этот пользователь должен существовать в системе; в противном случае агент не запускается. Если пользователь zabbix еще не существует, создайте его вместе с группой с таким же именем:

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Атрибуты login=false и rlogin=false не позволяют использовать учетную запись для входа в систему; агент по-прежнему может работать от имени этого пользователя.

Запуск агента вручную

Агент можно запустить вручную:

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Запуск агента с SRC

Регистрация агента

Агент также можно зарегистрировать в System Resource Controller (SRC). При запуске через SRC агент должен работать в режиме переднего плана (с параметром -f). В противном случае SRC потеряет контроль над процессом агента и не сможет его остановить. Например (выполните от имени пользователя root):

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

В этом примере:

- -s задаёт имя подсистемы, используемое с командами SRC;
- -p задаёт полный путь к двоичному файлу агента;
- -a задаёт аргументы командной строки агента в виде одной строки в кавычках; SRC разделяет эту строку на отдельные аргументы при запуске агента (принимается только один параметр -a);
- -u задаёт пользователя, от имени которого работает агент; имя пользователя преобразуется в числовой идентификатор пользователя при создании подсистемы, поэтому пользователь должен существовать заранее;
- -G добавляет агент в группу подсистемы tcpip, поэтому команды для этой группы также применяются к нему (например, stopsrc -g tcpip также останавливает агент);
- -S -n 15 -f 9 указывает SRC управлять агентом с помощью сигналов: stopsrc отправляет SIGTERM (15) для обычной остановки и SIGKILL (9) для принудительной остановки (stopsrc -f);
- -R указывает SRC перезапускать агент при его аварийном завершении (например, если происходит сбой одного из его процессов); после выполнения stopsrc агент не перезапускается.

Поскольку идентификатор пользователя сохраняется в определении подсистемы, обновите это определение, если пользователь zabbix создан заново с другим идентификатором пользователя:

```
chssys -s zabbix_agentd -u zabbix
```

Запуск и остановка агента

Агент можно запустить, проверить его состояние и остановить с помощью команд startsrc, lssrc и stopsrc:

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Не используйте принудительную остановку (stopsrc -f). Она завершает только основной процесс агента; остальные процессы агента продолжают работать и удерживают порт прослушивания открытым, поэтому агент нельзя запустить снова, пока эти процессы не будут завершены вручную (например, с помощью kill -9).

Запуск агента при загрузке системы

Регистрация агента в SRC не приводит к его запуску при загрузке системы. Чтобы агент автоматически запускался при загрузке, добавьте запись в /etc/inittab:

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Удаление агента из SRC

Чтобы удалить агент из автозапуска системы и из SRC:

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Агент 2

Обзор Zabbix agent 2 — это новое поколение агента Zabbix, написанное на Go (с повторным использованием части кода на C из Zabbix agent). Он предназначен для того, чтобы:

- уменьшить количество TCP-соединений;
- обеспечить улучшенную параллельность проверок;
- легко расширяться с помощью плагинов, которые предоставляют простые проверки с минимальным объемом кода и поддерживают сложные проверки, состоящие из долго выполняющихся сценариев и автономного сбора данных с периодической передачей результатов;
- служить заменой Zabbix agent, поддерживая все предыдущие возможности.

Все метрики в Zabbix agent 2 собираются плагинами.

Для плагинов характерны следующие особенности:

- Планируемые и гибкие интервалы как для пассивных, так и для активных проверок
- Управление очередью задач с учетом расписания и параллельного выполнения задач
- Тайм-ауты на уровне плагина
- Проверка совместимости Zabbix agent 2 и его плагинов при запуске

Существуют два типа плагинов:

- Встроенные плагины
- Загружаемые плагины

Вы также можете создавать собственные плагины.

Плагины, созданные пользователем, не обязательно должны быть встроены в agent 2. Их можно добавлять как загружаемые плагины, что упрощает создание плагинов для новых метрик.

Загружаемые плагины можно добавлять во время работы без перекомпиляции Zabbix agent 2.

См. также:

- Сравнение agent и agent 2
- Параметры конфигурации: Zabbix agent 2
- Параметры конфигурации: Zabbix agent 2 (Windows)
- Параметры конфигурации: Плагины
- Протокол плагина Zabbix agent 2

Пассивные и активные проверки

Агент Zabbix 2 поддерживает пассивные и активные проверки, как и агент Zabbix. Кроме того, активные проверки агента Zabbix 2 поддерживают гибкие интервалы/интервалы планирования и параллельное выполнение проверок в рамках одного активного сервера.

> **Note:**
> По умолчанию после перезапуска агент Zabbix 2 запланирует первый сбор данных для активных проверок на условно случайное время в пределах интервала обновления элемента данных, чтобы предотвратить скачки потребления ресурсов. Чтобы выполнять активные проверки, для которых не задан *Scheduling* интервал обновления, сразу после перезапуска агента, задайте параметр `ForceActiveChecksOnStart` (глобального уровня) или параметр плагина `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (влияет только на проверки определенного плагина) в файле конфигурации. Если параметр уровня плагина задан, он переопределяет глобальный параметр.

Параллелизм проверок

Проверки из разных плагинов могут выполняться параллельно. Количество параллельных проверок в пределах одного плагина ограничено настройкой производительности плагина. Каждый плагин может иметь жёстко зашитую в код настройку производительности (по умолчанию, 1000), значение которой можно уменьшить, указав настройку `Plugins.<  >.System.Capacity=N` в параметре конфигурации `Plugins`.

Поддерживаемые платформы Список поддерживаемых платформ см. на странице Требования.

Агент 2 в Unix-подобных системах Zabbix агент 2 в Unix-подобных системах запускается на узле сети, который находится под мониторингом.

Установка

Zabbix агент 2 можно установить в системах на базе Linux одним из следующих способов:

- Пакеты Zabbix - выберите компонент Agent 2 (после выбора версии Zabbix, дистрибутива ОС и версии ОС) и следуйте инструкциям;
- Исходные тексты Zabbix - загрузите исходные файлы и скомпилируйте агент, выполнив настройку с параметром `--enable-agent2`.

> **Note:**
> Возможности мониторинга Zabbix агент 2 можно расширить с помощью загружаемых плагинов, которые доступны отдельно. Подробности см. в разделе Загружаемые плагины.

Если установлен как пакет

Zabbix agent 2 работает как процесс переднего плана и для выполнения в фоновом режиме использует внешний менеджер служб (например, systemd). В Linux у Zabbix agent 2 нет встроенной поддержки демонизации.

Агент можно запустить, выполнив:

```
systemctl start zabbix-agent2
```

Чтобы остановить, перезапустить или проверить состояние Zabbix agent 2, используйте следующие команды:

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

Запуск вручную

Вы можете запустить агент Zabbix, найдя бинарный файл `zabbix_agent2` и запустив его напрямую; например:

```
zabbix_agent2
```

Агент 2 в системах Windows Zabbix агент 2 работает как отдельный процесс; однако его также можно запускать как службу Windows.

Установка

Zabbix agent 2 можно установить в Windows одним из следующих способов:

- Предварительно скомпилированные двоичные файлы Zabbix agent - загрузите MSI-пакет установщика агента и следуйте инструкциям на странице Установка агента в Windows из MSI;
- Исходные тексты Zabbix - загрузите исходные файлы и следуйте инструкциям на странице Сборка Zabbix agent 2 в Windows.

> **Note:**
> Возможности мониторинга Zabbix agent 2 можно расширить с помощью загружаемых плагинов, которые доступны отдельно. Подробности см. в разделе Загружаемые плагины.

Дополнительные сведения об установке Zabbix agent 2 (из ZIP-архива) в качестве службы Windows см. на странице Zabbix agent в Microsoft Windows.

Параметры С агентом Zabbix 2 можно использовать следующие параметры командной строки:

Параметр	Описание
Агент UNIX и Windows	
-c --config <config-file>	Путь к файлу конфигурации. Этот параметр можно использовать для указания файла конфигурации, отличного от файла по умолчанию.
-f --foreground	Запустить агент Zabbix в режиме переднего плана (по умолчанию: true).
-p --print	Вывести известные элементы данных и завершить работу. Чтобы также получить результаты пользовательских параметров, необходимо указать файл конфигурации (если он находится не в расположении по умолчанию).
-t --test <item key>	Проверить указанный элемент данных и завершить работу. Чтобы также получить результаты пользовательских параметров, необходимо указать файл конфигурации (если он находится не в расположении по умолчанию).
-T --test-config	Проверить файл конфигурации и завершить работу.
-h --help	Вывести справочную информацию и завершить работу.
-v --verbose	Вывести отладочную информацию. Используйте этот параметр вместе с параметрами -p и -t.
-V --version	Вывести информацию о версии агента и лицензии.
-R --runtime-control <option>	Выполнить административные функции. См. раздел управление во время выполнения.
Только агент Windows	
-m --multiple-agents	Использовать несколько экземпляров агента (с параметрами -i, -d, -s, -x). Чтобы различать имена служб экземпляров, имя каждой службы будет включать значение Hostname из указанного файла конфигурации.
-S --startup-type <value>	Установить тип запуска службы агента Zabbix для Windows. Допустимые значения: automatic - *(по умолчанию)* автоматически запускать службу при запуске Windows; delayed - отложить запуск службы до завершения запуска служб, запущенных автоматически; manual - запускать службу вручную (пользователем или приложением); disabled - отключить службу, чтобы ее нельзя было запустить пользователю или приложению. Этот параметр можно использовать вместе с параметром -i или отдельно для изменения типа запуска уже установленной службы.
-i --install	Установить агент Zabbix для Windows как службу.
-d --uninstall	Удалить службу агента Zabbix для Windows.
-s --start	Запустить службу агента Zabbix для Windows.
-x --stop	Остановить службу агента Zabbix для Windows.

Ниже приведены конкретные примеры использования параметров командной строки:

- Вывод всех встроенных элементов данных агента со значениями.
- Проверка пользовательского параметра с ключом mysql.ping, определенным в указанном файле конфигурации.
- Установка службы агента Zabbix для Windows с использованием пути по умолчанию к файлу конфигурации.
- Изменение типа запуска установленной службы агента Zabbix для Windows с использованием файла конфигурации, расположенного в той же папке, что и исполняемый файл агента.

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

Управление во время выполнения

Управление во время выполнения предоставляет несколько вариантов удалённого управления.

Option	Description
log_level_increase	Увеличить уровень журнала.
log_level_decrease	Уменьшить уровень журнала.
metrics	Вывести список доступных метрик.

Option	Description
`periodic_prof_enable`	Включить периодическое профилирование. Все метрики собирают данные в текущий момент и записывают их, за исключением данных о ЦП. Данные о ЦП периодически собираются в течение определённого времени во временный скрытый файл. По завершении интервала файл публикуется как профиль ЦП, при этом сохраняется настроенное количество завершённых файлов. Временный файл не учитывается в `ProfilerMaxFilesPerProfile`.
`periodic_prof_disable`	Отключить периодическое профилирование.
`periodic_prof_execute`	Записать профили после 5-секундного сбора данных. Если периодическое профилирование включено, эта команда завершает текущий интервал сбора данных о ЦП, записывает отдельный 5-секундный профиль ЦП и запускает новый периодический файл ЦП. Явно запрошенные профили ротируются в соответствии с `ProfilerMaxFilesPerProfile` после завершения команды. Пока эта команда собирает профили, все команды управления профилировщиком во время выполнения отклоняются с сообщением "profiler is busy".
`periodic_prof_set_interval`	Установить интервал профилирования.
`version`	Отобразить версию агента.
`userparameter_reload`	Повторно загрузить значения параметров `UserParameter` и `Include` из текущего файла конфигурации.
`help`	Отобразить справочную информацию об управлении во время выполнения.

Примеры:

- Увеличение уровня журнала для агента 2.
- Вывод вариантов управления во время выполнения.

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

Файл конфигурации Путь по умолчанию к файлу конфигурации зависит от операционной системы и способа установки Zabbix:

- В UNIX по умолчанию используется:
  - При установке из исходных кодов: `/usr/local/etc/zabbix_agent2.conf` или путь, заданный переменными на этапе компиляции `--sysconfdir` или `--prefix`.
  - При установке из пакетов: `/etc/zabbix/zabbix_agent2.conf`.
- В Windows по умолчанию используется `C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf`.

Если на узле сети запущен один экземпляр агента, он может использовать файл конфигурации по умолчанию или файл конфигурации, указанный в командной строке. При использовании нескольких экземпляров каждый экземпляр агента должен иметь собственный файл конфигурации (один из экземпляров может использовать файл конфигурации по умолчанию).

Параметры конфигурации агента Zabbix 2 в основном совпадают с параметрами агента Zabbix, за некоторыми исключениями:

Параметры	Описание
Параметры конфигурации, специфичные для агента Zabbix 2	
`ControlSocket`	Путь к управляющему сокету времени выполнения. Агент Zabbix 2 использует управляющий сокет для команд времени выполнения.
`EnablePersistentBuffer`, `PersistentBufferFile`, `PersistentBufferPeriod`	Эти параметры используются для настройки постоянного хранилища в агенте Zabbix 2 для активных элементов данных.

Параметры	Описание
ForceActiveChecksOnStart	Определяет, должен ли агент выполнять активные проверки сразу после перезапуска или распределять их равномерно во времени.
Plugins	У плагинов могут быть собственные параметры в формате Plugins.<Plugin name>.<Parameter>=<value>. Распространенным параметром плагина является System.Capacity, задающий ограничение на количество проверок, которые могут выполняться одновременно.
StatusPort	Порт, на котором агент Zabbix 2 будет прослушивать HTTP-запросы состояния и отображать список настроенных плагинов и некоторые внутренние параметры.
Параметры конфигурации агента Zab-bix, не поддерживаемые агентом Zab-bix 2	
AllowRoot, User	Не поддерживаются агентом Zabbix 2, поскольку он не поддерживает работу в режиме демона.
EnableRemoteCommands	Устарел; вместо него в обоих агентах используются параметры AllowKey, AllowKeyRegexp, DenyKey и DenyKeyRegexp.
ListenBacklog	Используется в агенте Zabbix для определения максимального количества ожидающих подключений в очереди TCP. Не поддерживается агентом Zabbix 2.
LoadModule, LoadModulePath	Загружаемые модули не поддерживаются агентом Zabbix 2.
MaxLinesPerSecond, LogRemoteCommands	В агенте Zabbix 2 вместо них используются Plugins.Log.MaxLinesPerSecond и Plugins.SystemRun.LogRemoteCommands.
StartAgents	Этот параметр используется в агенте Zabbix для увеличения количества одновременно выполняемых пассивных проверок или их отключения. В агенте Zabbix 2 количество одновременно выполняемых проверок настраивается на уровне плагина и может быть ограничено параметром емкости. Пассивные проверки отключены в агенте Zabbix 2, если параметр Server не указан.

Подробнее см. параметры файла конфигурации для агента Zabbix 2.

Коды выхода Zabbix agent 2 также может быть скомпилирован с более старыми версиями OpenSSL (1.0.1, 1.0.2).

В этом случае Zabbix предоставляет мьютексы для блокировки в OpenSSL. Если блокировка или разблокировка мьютекса завершается сбоем, то сообщение об ошибке выводится в стандартный поток ошибок (STDERR), а Agent 2 завершает работу с кодом возврата 2 или 3 соответственно.

Сравнение Agent и agent 2

В этом разделе описаны различия между Zabbix agent и Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Язык программирования	C	Go с некоторыми частями на C
Демонизация	yes	только через systemd (yes в Windows)
Поддерживаемые расширения	Пользовательские загружаемые модули на C.	Пользовательские плагины на Go.
Требования		
Поддерживаемые платформы	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: все настольные и серверные версии, начиная с XP	Linux, Windows: все настольные и серверные версии, на которые можно установить поддерживаемую версию Go.
Поддерживаемые криптографические библиотеки	GnuTLS 3.1.18 и новее. OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. LibreSSL - протестировано с версиями 2.7.4, 2.8.2 (применяются определенные ограничения, см. страницу Encryption для подробностей).	Linux: OpenSSL 1.0.1 и новее. MS Windows: OpenSSL 1.1.1 или новее. Библиотека OpenSSL должна иметь включенную поддержку PSK. LibreSSL не поддерживается.
Мониторинг процессов		
Процессы	Отдельный процесс активных проверок для каждой записи server/proxy.	Один процесс с автоматически создаваемыми потоками. Максимальное число потоков определяется переменной окружения `GOMAXPROCS`.
Метрики	**UNIX**: см. список поддерживаемых элементов данных. **Windows**: см. список дополнительных специфичных для Windows элементов данных.	**UNIX**: все метрики, поддерживаемые Zabbix agent. Кроме того, agent 2 предоставляет нативное для Zabbix решение мониторинга для: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd и других целей мониторинга - см. полный список специфичных для agent 2 элементов данных. **Windows**: все метрики, поддерживаемые Zabbix agent, а также проверки `net.tcp.service*` для HTTPS, LDAP. Кроме того, agent 2 предоставляет нативное для Zabbix решение мониторинга для: PostgreSQL, Redis.
Параллельность	Активные проверки для одного server выполняются последовательно.	Проверки из разных плагинов или несколько проверок внутри одного плагина могут выполняться параллельно.
Сторонние ловушки	no	yes
Дополнительные возможности		
Постоянное хранилище	no	yes
Постоянные файлы для метрик `log*[]`	yes (только в Unix)	no
Загрузка данных журнала	Может выполняться во время сбора журнала для освобождения буфера.	Сбор журнала останавливается, когда буфер заполнен, поэтому параметр `BufferSize` должен быть не меньше `MaxLinesPerSecond` x 2.
Изменение пользователя во время работы	yes (только Unix-подобные системы)	no (управляется systemd)
Настраиваемые пользователем наборы шифров	yes	no

См. также:

- *Описание процессов Zabbix*: Zabbix agent, Zabbix agent 2
- *Параметры конфигурации*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Встроенные плагины

Эти плагины встроены в Zabbix agent 2 и доступны сразу после установки. Щелкните имя плагина, чтобы перейти в репозиторий плагина с дополнительной информацией.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии
Агент	Метрики используемого агента Zabbix.	agent.hostname, agent.ping, agent.version	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
CPU	Мониторинг CPU системы (количество CPU/ядер CPU, обнаруженные CPU, процент загрузки).	system.cpu.discovery, system.cpu.num, system.cpu.util	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Docker	Мониторинг контейнеров Docker.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	См. также: Параметры конфигурации
Файл	Сбор метрик файлов.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Kernel	Мониторинг ядра.	kernel.maxfiles, kernel.maxproc	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Log	Мониторинг файлов журналов.	log, log.count, logrt, logrt.count	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys. См. также: Параметры конфигурации плагина (Unix/Windows)
Memcached	Мониторинг сервера Memcached.	memcached.ping, memcached.stats	
Modbus	Чтение данных Modbus.	modbus.get	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
MQTT	Получение опубликованных значений тем MQTT.	mqtt.get	Чтобы настроить зашифрованное соединение с брокером MQTT, укажите параметры TLS в файле конфигурации агента как параметры именованной сессии или значения по умолчанию. В настоящее время параметры TLS нельзя передавать как параметры ключа элемента данных.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии
MySQL	Мониторинг MySQL и его форков.	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	Чтобы настроить зашифрованное соединение с базой данных, укажите параметры TLS в файле конфигурации агента как параметры именованной сессии или значения по умолчанию. В настоящее время параметры TLS нельзя передавать как параметры ключа элемента данных.
NetIf	Мониторинг сетевых интерфейсов.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Oracle	Мониторинг Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Перед использованием плагина установите Oracle Instant Client.
Proc	Процент загрузки CPU процесса.	proc.cpu.util	Поддерживаемый ключ имеет те же параметры, что и ключ агента Zabbix key.
Redis	Мониторинг сервера Redis.	redis.config, redis.info, redis.ping, redis.slowlog.count	Чтобы настроить зашифрованное соединение с Redis, укажите параметры TLS в файле конфигурации агента как параметры именованной сессии или значения по умолчанию. Параметры TLS нельзя передавать как параметры ключа элемента данных. Обратите внимание, что некорректная или иным образом недопустимая конфигурация TLS может помешать запуску Zabbix agent 2, поэтому перед включением TLS проверьте файлы сертификатов, права доступа и пути.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии

Smart	Мониторинг S.M.A.R.T.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	Минимально необходимая версия smartctl — 7.1.
			Для пользователя, запускающего Zabbix agent 2, требуются права sudo/root для smartctl. Плагин использует только следующие команды:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Поддерживаемые keys можно использовать с Zabbix agent 2 только в Linux/Windows, как для пассивной, так и для активной проверки.
			См. также:
			Параметры конфигурации

SW	Список установленных пакетов.	system.sw.packages, system.sw.packages.get	Поддерживаемые ключи имеют те же параметры, что и ключ агента Zabbix key.
Swap	Размер пространства подкачки в байтах/процентах.	system.swap.size	Поддерживаемый ключ имеет те же параметры, что и ключ агента Zabbix key.
SystemRun	Выполняет указанный команду.	system.run	Поддерживаемый ключ имеет те же параметры, что и ключ агента Zabbix key. См. также: Параметры конфигурации плагина (Unix/Windows)
Systemd	Мониторинг служб systemd.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	Проверка доступности TCP-соединения.	net.tcp.port	Поддерживаемый ключ имеет те же параметры, что и ключ агента Zabbix key.
UDP	Мониторинг доступности и производительности служб UDP.	net.udp.service, net.udp.service.perf	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Uname	Получение информации о системе.	system.hostname, system.sw.arch, system.uname	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
Uptime	Сбор метрик времени работы системы.	system.uptime	Поддерживаемый ключ имеет те же параметры, что и ключ агента Zabbix key.
VFSDev	Сбор метрик VFS.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
WebCertificate	Мониторинг TLS/SSL-сертификатов веб-сайтов.	web.certificate.get	
WebPage	Мониторинг веб-страниц.	web.page.get, web.page.perf, web.page.regexp	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии
ZabbixAsync	Асинхронный сбор метрик.	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
ZabbixStats	Внутренние метрики сервера/прокси Zabbix или количество задержанных элементов данных в очереди.	zabbix.stats	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.
ZabbixSync	Синхронный сбор метрик.	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	Поддерживаемые ключи имеют те же параметры, что и ключи агента Zabbix keys.

Загружаемые плагины

Загружаемые плагины не входят в стандартную поставку агента 2 Zabbix и должны устанавливаться отдельно:

- В Linux можно использовать пакеты (например, установить Ember+ в Ubuntu с помощью `apt install zabbix-agent2-plugi` или собрать плагины
- В Windows можно установить плагины из MSI или собрать плагины

Attention:
Перед установкой плагина ознакомьтесь с его файлом README. В нем могут содержаться особые требования и инструкции по установке.

Нажмите имя плагина, чтобы перейти в репозиторий плагина, содержащий файл README с дополнительной информацией.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии
Ceph	Мониторинг Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	См. также параметры конфигурации плагина Ceph.
Ember+	Мониторинг Ember+.	ember.get	См. также параметры конфигурации плагина Ember+.
MongoDB	Мониторинг серверов и кластеров MongoDB (распределенной документно-ориентированной базы данных).	mongodb.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	Чтобы настроить зашифрованные подключения к базе данных, укажите параметры TLS в файле конфигурации агента в качестве параметров именованного сеанса. В настоящее время параметры TLS нельзя передавать в качестве параметров ключа элемента данных. См. также параметры конфигурации плагина MongoDB.
MSSQL	Мониторинг базы данных MSSQL.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	Чтобы настроить зашифрованное подключение к базе данных, укажите параметры TLS в файле конфигурации агента в качестве параметров именованного сеанса или параметров по умолчанию. В настоящее время параметры TLS нельзя передавать в качестве параметров ключа элемента данных. См. также параметры конфигурации плагина MSSQL.

Имя плагина	Описание	Поддерживаемые ключи элементов данных	Комментарии
NVIDIA GPU	Мониторинг GPU NVIDIA.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	См. также параметры конфигурации плагина NVIDIA GPU.
PostgreSQL	Мониторинг PostgreSQL и его форков.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	Чтобы настроить зашифрованные подключения к базе данных, укажите параметры TLS в файле конфигурации агента в качестве параметров именованного сеанса или параметров по умолчанию. В настоящее время параметры TLS нельзя передавать в качестве параметров ключа элемента данных. См. также параметры конфигурации плагина PostgreSQL.

Note:
Загружаемые плагины, запущенные со следующими параметрами:
 - `-V --version` - выводят сведения о версии плагина и лицензии;
 - `-h --help` - выводят справочную информацию.
 - `-t, --test <item key>` - запускают плагин для тестирования (конфигурация плагина игнорируется).

Параметры конфигурации: Zabbix агент 2

Обзор

Параметры конфигурации позволяют настраивать поведение агента 2 Zabbix.

Агент 2 Zabbix — это агент Zabbix нового поколения, который можно использовать вместо агента Zabbix.

Укажите значения параметров в файле конфигурации агента 2 Zabbix (`zabbix_agent2.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр является обязательным.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть полную информацию о параметре, нажмите его название)
- Подробная информация о параметрах

Параметр	Описание
Alias	Задать псевдоним для ключа элемента данных.
AllowKey	Разрешить выполнение ключей элементов данных, соответствующих шаблону.
AllowKeyRegexp	Разрешить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
BufferSend	Задать частоту отправки значений из буфера на сервер/прокси Zabbix.
BufferSize	Задать максимальное количество значений в буфере памяти.
ControlSocket	Указать управляющий сокет для отправки команд во время выполнения с параметром '-R'.
DebugLevel	Задать уровень отладки.
DenyKey	Запретить выполнение ключей элементов данных, соответствующих шаблону.
DenyKeyRegexp	Запретить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
EnablePersistentBuffer	Включить использование локального постоянного хранилища для активных элементов данных.
EnableProfiler	Включить периодическое профилирование Go при запуске агента.
ForceActiveChecksOnStart	Выполнить активные проверки сразу после перезапуска для первой полученной конфигурации.
HeartbeatFrequency	Задать частоту сообщений проверки доступности в секундах.
HostInterface	Указать интерфейс узла сети.
HostInterfaceItem	Указать элемент данных для получения интерфейса узла сети.
HostMetadata	Указать метаданные узла сети.
HostMetadataItem	Указать элемент данных агента Zabbix для получения метаданных узла сети.
Hostname	Указать имя узла сети.
HostnameItem	Указать элемент данных агента Zabbix для получения имени узла сети.
Include	Указать каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
ListenIP	Указать IP-адрес агента, на котором он ожидает подключения от сервера/прокси.
ListenPort	Указать порт, на котором ожидает подключения агент.
LogFile	Указать файл журнала агента.
LogFileSize	Задать максимальный размер файла журнала.
LogType	Указать тип вывода журнала.
PersistentBufferFile	Указать файл для хранения базы данных SQLite.
PersistentBufferPeriod	Задать период хранения данных при отсутствии подключения к серверу или прокси Zabbix.
PidFile	Указать PID-файл.
Plugins.<PluginName>.System.Capacity	Задать ограничение на количество проверок для каждого подключаемого модуля, которые могут выполняться одновременно.
Plugins.Log.MaxLinesPerSecond	Задать максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок `log` или `logrt`.
Plugins.SystemRun.LogRemoteCommands	Включить запись выполняемых команд оболочки в журнал как предупреждений.
PluginSocket	Указать путь к сокету UNIX для взаимодействия с загружаемыми подключаемыми модулями.
PluginTimeout	Задать время ожидания подключений к загружаемым подключаемым модулям.
ProfilerDir	Каталог, в который будут записываться данные профилирования.
ProfilerMaxFilesPerProfile	Максимальное количество файлов, сохраняемых для каждого типа профиля.
ProfilerInterval	Интервал записи данных профилирования.
RefreshActiveChecks	Указать частоту обновления списка активных проверок.
Server	Указать IP-адреса или DNS-имена серверов/прокси Zabbix.

Параметр	Описание
ServerActive	Указать адрес сервера/прокси Zabbix или конфигурацию кластера, откуда будут получаться активные проверки.
SourceIP	Указать исходный IP-адрес.
StatusPort	Указать порт для прослушивания запросов состояния HTTP (`http://localhost:<port>/status`).
Timeout	Указать, как долго (в секундах) ожидать установления соединения и обмена данными с сервером или прокси Zabbix.
TLSAccept	Указать требуемый уровень шифрования для входящих подключений.
TLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации для проверки сертификатов узлов, используемые для зашифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Указать полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемые для зашифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll	Указать строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Указать строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или более поздней версии. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Указать строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Указать строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или более поздней версии. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Указать строку шифров OpenSSL (TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Указать строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или более поздней версии. Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Указать требуемый уровень шифрования для подключений к серверу или прокси Zabbix.
TLSCRLFile	Указать полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованного взаимодействия между компонентами Zabbix.
TLSPSKFile	Указать полный путь к файлу, содержащему предварительно согласованный ключ агента, используемый для зашифрованного взаимодействия с сервером или прокси Zabbix.
TLSPSKIdentity	Указать строку идентификатора предварительно согласованного ключа, используемую для зашифрованного взаимодействия с сервером или прокси Zabbix.
TLSServerCertIssuer	Указать разрешенного издателя сертификата сервера (прокси).
TLSServerCertSubject	Указать разрешенную тему сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу всех символов в аргументах пользовательских параметров.
UserParameter	Указать пользовательский параметр для мониторинга.
UserParameterDir	Указать путь поиска по умолчанию для команд UserParameter.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

Псевдоним

Задайте псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Можно указать несколько параметров `Alias`. Несколько параметров с одинаковым ключом `Alias` не допускаются.
 Разные ключи `Alias` могут ссылаться на один и тот же ключ элемента данных.
 Псевдонимы можно использовать в параметре `HostMetadataItem`, но не в параметре `HostnameItem`.

Пример 1: получение идентификатора пользователя `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Теперь для получения данных можно использовать сокращенный ключ zabbix.userid.

Пример 2: получение загрузки ЦП с параметрами по умолчанию и пользовательскими параметрами.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Это позволяет использовать ключ cpu.util для получения процента загрузки ЦП с параметрами по умолчанию, а также cpu.util[all, idle, avg15] для получения определенных данных о загрузке ЦП.

Пример 3: запуск нескольких правил обнаружения низкоуровневых объектов, обрабатывающих одни и те же элементы данных обнаружения.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Теперь можно настроить несколько правил обнаружения, используя vfs.fs.discovery с разными параметрами для каждого правила, например vfs.fs.discovery[foo], vfs.fs.discovery[bar] и т. д.

AllowKey

Разрешить выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа — это выражение с подстановочными знаками, которое поддерживает символ * для сопоставления с любым количеством любых символов.
Несколько правил сопоставления ключей могут быть определены в сочетании с DenyKey. Несколько правил сопоставления ключей могут быть определены в сочетании с DenyKey, AllowKeyRegexp и DenyKeyRegexp. Параметры обрабатываются по одному в порядке их следования. Порядок определяет, какое из совпадающих правил будет иметь приоритет. См. также: Ограничение проверок агента.

AllowKeyRegexp

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с DenyKeyRegexp, AllowKey и DenyKey. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

Пример:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Установите частоту отправки значений из буфера на сервер/прокси Zabbix (в секундах).
 Обратите внимание, что если буфер заполнен, данные будут отправлены раньше.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Установите максимальное количество значений в буфере памяти.
 Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнится. Этот параметр следует использовать только если постоянный буфер отключен (EnablePersistentBuffer=0).

По умолчанию: 1000
 Диапазон: 2-65535

ControlSocket

Укажите управляющий сокет для отправки команд во время работы с помощью параметра -R.

По умолчанию: /tmp/agent.sock

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

DenyKey

Запретить выполнение тех элементов данных, ключи которых соответствуют шаблону.
 Шаблон ключа представляет собой выражение с подстановочными символами, поддерживающее символ * для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с AllowKey. Можно

определить несколько правил сопоставления ключей в сочетании с AllowKey, AllowKeyRegexp и DenyKeyRegexp. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

По умолчанию: DenyKey=system.run[*]

DenyKeyRegexp

Запретить выполнение тех ключей элементов данных, которые соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с AllowKeyRegexp, AllowKey и DenyKey. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет.
 Все элементы данных system.run[] по умолчанию запрещены (DenyKey=system.run[*]), если не указано другое правило system.run[*].

См. также: Ограничение проверок агентом.

Пример:

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

Включить использование локального постоянного хранилища для активных элементов данных.
 Если постоянное хранилище отключено, будет использоваться буфер памяти.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

EnableProfiler

Включает периодическое профилирование Go при запуске агента. Профилирование также можно включить или отключить во время выполнения с помощью параметра -R. Пока periodic_prof_execute собирает профили, все команды профилировщика во время выполнения отклоняются с сообщением о занятости.

Значение по умолчанию: 0
 Значения: 0 - отключено, 1 - включено

ForceActiveChecksOnStart

Выполнять активные проверки сразу после перезапуска для первой полученной конфигурации.
 Также доступно как параметр конфигурации для каждого плагина, например: Plugins.Uptime.System.ForceActiveChecksOnStart=1

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

HeartbeatFrequency

Установите частоту сообщений heartbeat в секундах.
 Используется для мониторинга доступности активных проверок.
0 - сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Укажите интерфейс узла сети (IP-адрес или DNS-имя), используемый во время авторегистрации узла сети.
 Это значение будет использовано для заполнения интерфейса в вновь созданном узле сети и позволяет явно задать либо IP-адрес, либо DNS-адрес.
 Подробнее см. Использование DNS в качестве интерфейса по умолчанию.

Если не задано, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает лимит в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Укажите элемент данных для определения интерфейса узла сети (IP-адреса или DNS-имени) во время автоматической регистрации узла сети.
 Это значение используется только в том случае, если параметр HostInterface не определен.
 Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса на автоматическую регистрацию агент зарегистрирует предупреждение, если значение, возвращенное указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

HostMetadata

Укажите метаданные, используемые для идентификации или различения узла сети во время автоматической регистрации узла сети.
 HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если параметр не задан, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или не является строкой в кодировке UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, корректные в UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечён на первом переносе строки.

Диапазон: 0–2034 байта

HostMetadataItem

Укажите элемент данных для получения метаданных узла сети.
 Этот параметр используется только в том случае, если HostMetadata не задан.

Значение HostMetadataItem извлекается при каждой попытке автоматической регистрации и используется только в процессе автоматической регистрации узла сети. HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Поддерживаются пользовательские параметры и псевдонимы. Элемент данных system.run[] поддерживается независимо от настроек AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

Во время запроса на автоматическую регистрацию агент записывает предупреждение в журнал, если значение, возвращённое указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращённое элементом данных, должно быть строкой в кодировке UTF-8; в противном случае оно игнорируется. Если параметр ожидает IP-адрес или DNS-имя, значения, корректные с точки зрения UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также отклоняются и помечаются как недопустимые.

Многострочные метаданные не поддерживаются — вывод усекается на первом переносе строки.

Имя узла сети

Укажите уникальные имена узлов сети с учетом регистра. Список, разделенный запятыми.
 Требуется для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере.
 Значение берется из HostnameItem, если оно не задано.
Допустимые символы: буквенно-цифровые, '.', ' ', '_' и '-'. Максимальная длина: 128 символов на каждое имя узла сети, 2048 символов для всей строки.

По умолчанию: задается через HostnameItem

Элемент данных Hostname

Укажите элемент данных для получения имени узла сети.
 Этот параметр используется только в том случае, если Hostname не определен.
 Пользовательские параметры и псевдонимы не поддерживаются, однако элемент данных system.run[] поддерживается независимо от значений AllowKey, AllowKeyRegexp, DenyKey или DenyKeyRegexp.

По умолчанию: system.hostname

Include

Укажите каталог или отдельные файлы, которые нужно включить в файл конфигурации.
 Во время установки Zabbix этот параметр устанавливается в /usr/local/etc, если вы не измените его на этапе компиляции.

Общие правила для параметра Include:

- Путь может быть относительным к расположению файла zabbix_agent2.conf.
- В одном файле конфигурации можно указать несколько параметров Include.
- Zabbix обрабатывает параметр Include сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила для включения каталогов или шаблонов файлов:

- Не следует предполагать какой-либо порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общее значение более конкретным), так как невозможно предсказать, какое значение будет использовано.
- Обратите внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании include/my_specific.conf может быть создан файл include/my_specific.conf.BAK), поскольку будут включены оба файла.

Чтобы включать только нужные файлы из каталога, используйте подстановочный знак (*) для сопоставления шаблонов.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Укажите IP-адрес, на котором агент принимает подключения от сервера/прокси. Список разделяется запятыми.
 Первый IP-адрес отправляется на сервер Zabbix, если подключение выполняется к нему, чтобы получить список активных проверок.

По умолчанию: 0.0.0.0

ListenPort

Укажите порт прослушивания агента для подключений от сервера/прокси.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Укажите файл журнала агента.

По умолчанию: /tmp/zabbix_agent2.log
 Обязательно: Да, если LogType установлено в file; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: Если достигается ограничение размера файла журнала и ротация файла по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *system* - записывать журнал в syslog; - *console* - записывать журнал в стандартный вывод.

По умолчанию: file

PersistentBufferFile

Укажите файл для хранения базы данных SQLite. Должно быть указано полное имя файла.
 Этот параметр используется только если включен постоянный буфер (EnablePersistentBuffer=1).

PersistentBufferPeriod

Задайте период времени для хранения данных при отсутствии соединения с сервером Zabbix или прокси. Более старые данные будут потеряны.
 Данные журнала будут сохранены. Этот параметр используется только если включен постоянный буфер (EnablePersistentBuffer=1).

По умолчанию: 1h
 Диапазон: 1m-365d

PidFile

Укажите PID-файл.

По умолчанию: /tmp/zabbix_agent2.pid

Плагины.<PluginName>.System.Capacity

Установите ограничение на количество проверок для плагина <PluginName>, которые могут выполняться одновременно.

По умолчанию: 1000 Диапазон: 1-1000

Plugins.Log.MaxLinesPerSecond

Установите максимальное количество новых строк в секунду для отправки на сервер/прокси Zabbix при обработке активных проверок log или logrt.
 Указанное значение будет переопределено параметром maxlines, заданным в ключе элемента данных log и logrt.
Примечание: Zabbix будет обрабатывать в 10 раз больше новых строк, чем задано в MaxLinesPerSecond, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

Plugins.SystemRun.LogRemoteCommands

Включить запись выполненных команд оболочки в журнал как предупреждения.
 Команды будут записываться в журнал только при удаленном выполнении.
 Записи в журнале не будут создаваться, если `system.run[]` запускается локально параметрами `HostMetadataItem`, `HostInterfaceItem` или `HostnameItem`.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

PluginSocket

Укажите путь к UNIX-сокету для связи загружаемых плагинов.

По умолчанию: `/tmp/agent.plugin.sock`

PluginTimeout

Укажите тайм-аут для подключений к загружаемым плагинам в секундах.

По умолчанию: `Timeout`
 Диапазон: 1-30

ProfilerDir

Каталог, в который будут записываться данные профилирования.

Значение по умолчанию: `/tmp/zabbix_agent2_profiles`

ProfilerMaxFilesPerProfile

Максимальное количество файлов, сохраняемых для каждого типа профиля. Ограничение применяется к завершённым файлам после каждого дампа профиля. Активный профиль ЦП собирается в скрытый временный файл и не учитывается при расчёте ограничения. На файлы, созданные с помощью `periodic_prof_execute`, также распространяется это ограничение.

Значение по умолчанию: 10
 Диапазон: 1-100

ProfilerInterval

Интервал записи данных профилирования в секундах.

Значение по умолчанию: 3600
 Диапазон: 1–86400

RefreshActiveChecks

Укажите частоту обновления списка активных проверок в секундах.
 Обратите внимание, что после неудачной попытки обновить активные проверки следующая попытка обновления будет выполнена через 60 секунд.

По умолчанию: 5
 Диапазон: 1-86400

Сервер

Укажите IP-адреса, при необходимости в нотации CIDR, или DNS-имена серверов/прокси Zabbix. Список, разделенный запятыми.
 Входящие соединения будут приниматься только от узлов сети, перечисленных здесь. Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются эквивалентными, а `::/0` позволит использовать любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес. Пробелы допускаются. Если этот параметр не указан, пассивные проверки будут отключены, и агент не будет прослушивать ни один TCP-порт.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Укажите адрес сервера/прокси Zabbix или конфигурацию кластера, чтобы получать активные проверки.
 Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделенные двоеточием.
 Конфигурация кластера — это один или несколько адресов членов группы серверов или прокси, разделенных точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделяя их запятыми. Если не используются группы прокси, для каждого сервера/кластера Zabbix не следует указывать более одного прокси Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделенных запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Допускаются пробелы.
Если порт не указан, используется порт по умолчанию.
IPv6-адреса должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для IPv6-адресов необязательны.
Если этот параметр не указан, активные проверки отключены.

Пример для прокси Zabbix:

```
ServerActive=127.0.0.1:10051
```

Пример для группы прокси Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Пример для нескольких серверов:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Пример для высокой доступности:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Пример для высокой доступности с двумя кластерами и одним сервером:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Укажите исходящий IP-адрес для:

- исходящих подключений к серверу Zabbix или прокси Zabbix.
- установления соединений при выполнении некоторых элементов данных (`web.page.get`, `net.tcp.port` и т. д.).

StatusPort

Укажите порт для прослушивания HTTP-запросов состояния (`http://localhost:<port>/status`).

Диапазон: 1024-32767

Таймаут

Укажите, сколько времени ожидать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций связи, включая:

- ожидание ответа от сервера Zabbix;
- отправку запросов на сервер Zabbix, включая запросы конфигурации элементов данных и данные элементов данных в активных проверках;
- получение данных журнала через logfile;
- отправку сообщений heartbeat;
- максимальную длительность проверок `vfs.*`;
- использование в качестве резервного значения в сценариях, когда сервер или прокси старше версии 7.0 отправляет проверки без таймаутов.

Этот таймаут **не** будет использоваться для тех проверок агента, для которых на веб-интерфейсе можно настроить значение таймаута (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Укажите уровень шифрования, требуемый для входящих подключений.
 Используется для пассивных проверок.
 Можно указать несколько значений, разделяя их запятыми: - *unencrypted* - принимать подключения без шифрования (по умолчанию); - *psk* - принимать подключения с TLS и предварительно общим ключом (PSK); - *cert* - принимать подключения с TLS и сертификатом.

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого центра сертификации (CA) для проверки сертификата узла-пира, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованного обмена данными с компонентами Zabbix.

TLSCipherAll

Укажите строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Укажите строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или новее. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Укажите строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Пример:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Обратите внимание, что этот параметр нельзя использовать вместе с `TLSAccept=cert,psk`; для подключения по сертификату (`TLSConnect=cert`) используйте вместо него `TLSCipherAll`.

TLSCipherCert13

Укажите строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или более новой версии. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Обратите внимание, что этот параметр нельзя использовать вместе с `TLSAccept=cert,psk`; для подключения по сертификату (`TLSConnect=cert`) используйте вместо него `TLSCipherAll13`.

TLSCipherPSK

Укажите строку шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Укажите строку шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или более новой версии. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Укажите уровень шифрования, требуемый для подключений агента к Zabbix server или proxy.
 Используется для активных проверок.
 Можно указать только одно значение: - *unencrypted* - подключение без шифрования (по умолчанию); - *psk* - подключение с использованием TLS и предварительно общего ключа (PSK); - *cert* - подключение с использованием TLS и сертификата.

Обязательно: да, если заданы параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCRLFile

Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованной связи между компонентами Zabbix.

TLSPSKFile

Укажите полный путь к файлу, содержащему предварительно общий ключ агента, используемый для зашифрованной связи с сервером или прокси Zabbix.

TLSPSKIdentity

Укажите строку идентификатора предварительно разделенного ключа, используемую для зашифрованной связи с сервером Zabbix или прокси.

TLSServerCertIssuer

Укажите допустимого издателя сертификата сервера (прокси).

TLSServerCertSubject

Укажите допустимый subject сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передачу всех символов в аргументах пользовательских параметров. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не допускаются.

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

UserParameter

Укажите пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: `UserParameter=<key>,<shell command>`
Обратите внимание, что команда оболочки не должна возвращать пустую строку или только EOL. Команды оболочки могут использовать относительные пути, если указан параметр `UserParameterDir`.

Пример:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Укажите путь поиска по умолчанию для команд `UserParameter`. Если этот параметр используется, агент изменит свой рабочий каталог на указанный здесь перед выполнением команды. Таким образом, команды UserParameter могут использовать относительный префикс `./` вместо полного пути.
Допускается только одна запись.

Пример:

```
UserParameterDir=/opt/myscripts
```

Параметры конфигурации: Zabbix agent 2 (Windows)

Обзор

Параметры конфигурации позволяют настраивать поведение агента Zabbix 2 в Windows.

Укажите значения параметров в файле конфигурации агента Zabbix 2 (`zabbix_agent2.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр является обязательным.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробную информацию о параметре, нажмите его имя)
- Подробная информация о параметрах

Параметр	Описание
Alias	Задать псевдоним для ключа элемента данных.
AllowKey	Разрешить выполнение ключей элементов данных, соответствующих шаблону.
AllowKeyRegexp	Разрешить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
BufferSend	Задать частоту отправки значений из буфера на сервер Zabbix.
BufferSize	Задать максимальное количество значений в буфере памяти.
ControlSocket	Указать управляющий сокет, используемый для отправки команд во время выполнения с параметром `-R`.
DebugLevel	Задать уровень отладки.
DenyKey	Запретить выполнение ключей элементов данных, соответствующих шаблону.
DenyKeyRegexp	Запретить выполнение ключей элементов данных, соответствующих шаблону на основе регулярного выражения.
EnablePersistentBuffer	Включить использование локального постоянного хранилища для активных элементов данных.
EnableProfiler	Включить периодическое профилирование Go при запуске агента.
ForceActiveChecksOnStart	Выполнить активные проверки сразу после перезапуска для первой полученной конфигурации.
HeartbeatFrequency	Задать частоту сообщений проверки доступности в секундах.
HostInterface	Указать интерфейс узла сети.
HostInterfaceItem	Указать элемент данных для получения интерфейса узла сети.

Параметр	Описание
HostMetadata	Указать метаданные узла сети.
HostMetadataItem	Указать элемент данных агента Zabbix, используемый для получения метаданных узла сети.
Hostname	Указать имя узла сети.
HostnameItem	Указать элемент данных агента Zabbix для получения имени узла сети.
Include	Указать каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
ListenIP	Указать IP-адрес прослушивания агента для подключений от сервера/прокси.
ListenPort	Указать порт прослушивания агента.
LogFile	Указать файл журнала агента.
LogFileSize	Задать максимальный размер файла журнала.
LogType	Указать тип вывода журнала.
PerfCounter	Указать новый параметр <parameter_name>, представляющий собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах).
PerfCounterEn	Указать новый параметр <parameter_name>, представляющий собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах). В отличие от PerfCounter, пути к счетчикам производительности должны быть указаны на английском языке.
PersistentBufferFile	Указать файл, в котором агент Zabbix 2 должен хранить базу данных SQLite.
PersistentBufferPeriod	Задать продолжительность хранения данных при отсутствии подключения к серверу или прокси.
Plugins.<PluginName>.System.Capacity	Задать ограничение на количество проверок для каждого плагина, которые могут выполняться одновременно.
Plugins.Log.MaxLinesPerSecond	Задать максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок `log`, `logrt` и `eventlog`.
Plugins.SystemRun.LogRemoteCommands	Включить запись выполненных команд оболочки в журнал как предупреждений.
PluginSocket	Указать путь к именованному каналу Windows для обмена данными с загружаемыми плагинами.
PluginTimeout	Задать время ожидания подключений к загружаемым плагинам в секундах.
ProfilerDir	Каталог, в который будут записываться данные профилирования.
ProfilerMaxFilesPerProfile	Максимальное количество файлов, сохраняемых для каждого типа профиля.
ProfilerInterval	Интервал записи данных профилирования.
RefreshActiveChecks	Указать частоту обновления списка активных проверок.
Server	Указать IP-адреса или DNS-имена серверов/прокси Zabbix.
ServerActive	Указать адрес сервера/прокси Zabbix или конфигурацию кластера, откуда необходимо получать активные проверки.
SourceIP	Указать исходный IP-адрес.
StatusPort	Задать порт, который агент прослушивает для HTTP-запросов состояния (`http://localhost:<port>/status`).
Timeout	Указать, как долго (в секундах) ожидать установления соединения и обмена данными с сервером или прокси Zabbix.
TLSAccept	Указать требуемый уровень шифрования для входящих подключений.
TLSCAFile	Указать полный путь к файлу, содержащему сертификаты корневого центра сертификации для проверки сертификатов узлов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCertFile	Указать полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемые для зашифрованного обмена данными между компонентами Zabbix.
TLSCipherAll13	Указать строку шифров OpenSSL (TLS 1.3). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert13	Указать строку шифров OpenSSL (TLS 1.3). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK13	Указать строку шифров OpenSSL (TLS 1.3). Переопределить критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Указать требуемый уровень шифрования для подключений к серверу или прокси Zabbix.
TLSCRLFile	Указать полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованного обмена данными между компонентами Zabbix.
TLSKeyFile	Указать полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованного обмена данными между компонентами Zabbix.
TLSPSKFile	Указать полный путь к файлу, содержащему предварительно установленный ключ агента, используемый для зашифрованного обмена данными с сервером или прокси Zabbix.

Параметр	Описание
TLSPSKIdentity	Указать строку идентификатора предварительно установленного ключа, используемую для зашифрованного обмена данными с сервером или прокси Zabbix.
TLSServerCertIssuer	Указать разрешенного издателя сертификата сервера (прокси).
TLSServerCertSubject	Указать разрешенное имя субъекта сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу всех символов в аргументах пользовательских параметров.
UserParameter	Указать пользовательский параметр для мониторинга.
UserParameterDir	Указать путь поиска по умолчанию для команд `UserParameter`.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные окружения;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

Псевдоним

Задайте псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Можно указать несколько параметров `Alias`. Несколько параметров с одинаковым ключом `Alias` не допускаются.
 Разные ключи `Alias` могут ссылаться на один и тот же ключ элемента данных.
 Псевдонимы можно использовать в `HostMetadataItem`, но не в параметре `HostnameItem`.

Пример 1: получение данных об использовании файла подкачки в процентах с сервера.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Теперь для получения данных можно использовать сокращенный ключ `pg_usage`.

Пример 2: получение загрузки ЦП с параметрами по умолчанию и пользовательскими параметрами.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Это позволяет использовать ключ `cpu.load` для получения загрузки ЦП с параметрами по умолчанию, а также `cpu.load[percpu,avg15]` для получения определенных данных о загрузке ЦП.

Пример 3: запуск нескольких правил обнаружения низкого уровня, обрабатывающих одни и те же элементы данных обнаружения.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Теперь можно настроить несколько правил обнаружения, используя `vfs.fs.discovery` с разными параметрами для каждого правила, например `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` и т. д.

AllowKey

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными знаками, которое поддерживает символ * для сопоставления с любым количеством любых символов.
 Можно определить несколько правил сопоставления ключей в сочетании с `DenyKey`, `AllowKeyRegexp` и `DenyKeyRegexp`. Параметры обрабатываются по одному в порядке их следования. Порядок определяет, какое из совпадающих правил будет иметь приоритет. См. также: Ограничение проверок агента.

AllowKeyRegexp

Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с `DenyKeyRegexp`, `AllowKey` и `DenyKey`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет. См. также: Ограничение проверок агента.

Пример:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Установите частоту отправки значений из буфера на сервер/прокси Zabbix (в секундах).
Обратите внимание: если буфер заполнен, данные будут отправлены раньше.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Установите максимальное количество значений в буфере памяти.
 Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнится.
Этот параметр следует использовать только в том случае, если постоянный буфер отключен (`EnablePersistentBuffer=0`).

По умолчанию: 1000
 Диапазон: 2-65535

ControlSocket

Укажите управляющий сокет, используемый для отправки команд во время работы с помощью параметра `-R`.

По умолчанию: `\\.\pipe\agent.sock`

DebugLevel

Установите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

DenyKey

Запретить выполнение тех ключей элементов данных, которые соответствуют шаблону.

Шаблон ключа представляет собой выражение с подстановочными символами и поддерживает символ `*`, соответствующий любому количеству любых символов.
 Можно определить несколько правил сопоставления ключей в сочетании с `AllowKey`, `AllowKeyRegexp` и `DenyKeyRegexp`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет.

См. также: Ограничение проверок агента.

По умолчанию: `DenyKey=system.run[*]`

DenyKeyRegexp

Запретить выполнение тех элементов данных, ключи которых соответствуют шаблону. Шаблон ключа представляет собой регулярное выражение.
 Можно определить несколько правил сопоставления ключей в сочетании с `AllowKeyRegexp`, `AllowKey` и `DenyKey`. Параметры обрабатываются по одному в порядке их появления. Порядок определяет, какое из совпавших правил будет иметь приоритет.
 Все элементы данных `system.run[]` по умолчанию запрещены (`DenyKey=system.run[*]`), если не указано другое правило `system.run[*]`.

См. также: Ограничение проверок агентом.

EnablePersistentBuffer

Включить использование локального постоянного хранилища для активных элементов данных. Если постоянное хранилище отключено, будет использоваться буфер памяти.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

EnableProfiler

Включает периодическое профилирование Go при запуске агента. Профилирование также можно включить или отключить во время выполнения с помощью параметра `-R`. Пока `periodic_prof_execute` собирает профили, все команды профилировщика во время выполнения отклоняются с сообщением о занятости.

Значение по умолчанию: 0
 Значения: 0 - отключено, 1 - включено

ForceActiveChecksOnStart

Выполнять активные проверки сразу после перезапуска для первой полученной конфигурации. Также доступно как параметр конфигурации для каждого плагина, например: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

HeartbeatFrequency

Установите частоту сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 - сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Укажите интерфейс узла сети (IP-адрес или DNS-имя), используемый во время авторегистрации узла сети. Это значение будет использовано для заполнения интерфейса в вновь созданном узле сети и позволяет явно указать либо IP-, либо DNS-адрес. Подробнее см. Использование DNS в качестве интерфейса по умолчанию.

Если не задано, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает лимит в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Укажите элемент данных для определения интерфейса узла сети (IP-адреса или DNS-имени) во время автоматической регистрации узла сети. Это значение используется только в том случае, если параметр `HostInterface` не определён. Дополнительные сведения см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса на автоматическую регистрацию агент зарегистрирует предупреждение, если значение, возвращённое указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных `system.run[]` поддерживается независимо от настроек `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.

HostMetadata

Укажите метаданные, используемые для идентификации или различения узла сети во время автоматической регистрации узла сети (активный агент). `HostMetadata` позволяет различать узлы сети не только по имени узла сети.

Если параметр не задан, значение будет получено из `HostMetadataItem`.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или не является строкой в кодировке UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, являющиеся допустимыми строками UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечён на первом переносе строки.

Диапазон: 0–2034 байта

HostMetadataItem

Укажите элемент данных агента Zabbix для получения метаданных узла сети. Этот параметр используется только в том случае, если `HostMetadata` не задан.

Значение `HostMetadataItem` извлекается при каждой попытке автоматической регистрации и используется только в процессе автоматической регистрации узла сети (активный агент). `HostMetadataItem` позволяет различать узлы сети не только по имени узла сети.

Поддерживаются пользовательские параметры и псевдонимы. Элемент данных `system.run[]` поддерживается независимо от настроек `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.

Во время запроса на автоматическую регистрацию агент запишет предупреждение в журнал, если значение, возвращенное указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращенное элементом данных, должно быть строкой в кодировке UTF-8; в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, корректные с точки зрения UTF-8, но не являющиеся допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет усечен на первом переносе строки.

Имя узла сети

Укажите уникальные имена узлов сети с учетом регистра. Список, разделенный запятыми.
 Обязательно для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Значение берется из HostnameItem, если оно не задано.
 Допустимые символы: буквенно-цифровые, '.', ' ', '_' и '-'. Максимальная длина: 128 символов на каждое имя узла сети, 2048 символов для всей строки.

По умолчанию: задается через HostnameItem

HostnameItem

Укажите элемент данных для получения имени узла сети. Этот параметр используется только в том случае, если `Hostname` не определен. Пользовательские параметры и псевдонимы не поддерживаются, однако элемент данных `system.run[]` поддерживается независимо от значений `AllowKey`, `AllowKeyRegexp`, `DenyKey` или `DenyKeyRegexp`.

Значение по умолчанию: `system.hostname`

Include

Укажите каталог или отдельные файлы, которые необходимо включить в файл конфигурации (по умолчанию находится в C:\Program Files\Zabbix Agent 2, если агент Zabbix установлен с помощью пакетов установщика Windows MSI; находится в папке, указанной во время установки, если агент Zabbix установлен из zip-архива). Во время установки Zabbix этому параметру присваивается значение /usr/local/etc, если оно не было изменено во время компиляции.

Общие правила для параметра Include:

- Путь может быть относительным к расположению файла zabbix_agent2.conf (например, Include=.\zabbix_agent2.d\pl
- В одном файле конфигурации можно указать несколько параметров Include.
- Zabbix обрабатывает параметр Include сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила включения каталогов или шаблонов файлов:

- Не рассчитывайте на определенный порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общую настройку конкретной), поскольку невозможно предсказать, какое значение будет использовано.
- Обращайте внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании include/my_specific.conf может быть создан файл include/my_specific.conf.BAK), поскольку будут включены оба файла.

Чтобы включить из каталога только нужные файлы, используйте подстановочный знак (*) для сопоставления с шаблоном.

Пример:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Укажите IP-адрес, на котором агент принимает подключения от сервера/прокси. Список, разделенный запятыми.
 Первый IP-адрес отправляется на сервер Zabbix, если выполняется подключение к нему, чтобы получить список активных проверок.

По умолчанию: 0.0.0.0

ListenPort

Укажите порт прослушивания агента для подключений от сервера/прокси.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Укажите файл журнала агента.

По умолчанию: c:\zabbix_agent2.log
 Обязательно: Да, если LogType установлен в file; в противном случае — нет

LogFileSize

Установите максимальный размер файла журнала (в МБ).
0 - отключить автоматическую ротацию журнала.
Примечание: если достигается ограничение размера файла журнала и ротация файла по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogType

Укажите тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *console* - записывать журнал в стандартный вывод.

По умолчанию: file

PerfCounter

Определите новый параметр <parameter_name>, который представляет собой среднее значение счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах).
Синтаксис: <parameter_name>,”<perf_counte

Например, если вы хотите получать среднее количество прерываний процессора в секунду за последнюю минуту, вы можете определить новый параметр ”interrupts” следующим образом:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Обратите внимание на двойные кавычки вокруг пути к счетчику производительности. Имя параметра (interrupts) следует использовать как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения будут собираться каждую секунду.
Вы можете выполнить `typeperf -qx`, чтобы получить список всех счетчиков производительности, доступных в Windows.

PerfCounterEn

Определите новый параметр <parameter_name>, который является средним значением счетчика производительности системы <perf_counter_path> за указанный период времени <period> (в секундах). В отличие от PerfCounter, пути perfcounter должны быть на английском языке. Поддерживается только в **Windows Server 2008/Vista** и более поздних версиях.
Синтаксис: <parameter_name>,"<perf_counter_path>",<period>

Например, если вы хотите получать среднее число прерываний процессора в секунду за последнюю минуту, вы можете определить новый параметр "interrupts" следующим образом:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Обратите внимание на двойные кавычки вокруг пути к счетчику производительности. Имя параметра (interrupts) будет использоваться как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения будут сниматься каждую секунду.
Список английских строк можно найти, просмотрев следующий раздел реестра: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

PersistentBufferFile

Укажите файл, в котором агент Zabbix 2 должен хранить базу данных SQLite. Должно быть указано полное имя файла.
 Этот параметр используется только если включен постоянный буфер (`EnablePersistentBuffer=1`).

PersistentBufferPeriod

Установите длительность хранения данных при отсутствии соединения с сервером или прокси. Более старые данные будут потеряны. Данные журнала будут сохранены.
 Этот параметр используется только если включен постоянный буфер (`EnablePersistentBuffer=1`).

По умолчанию: `1h`
 Диапазон: 1m-365d

Плагины.<PluginName>.System.Capacity

Установите лимит проверок для плагина `<PluginName>`, которые могут выполняться одновременно.

По умолчанию: 1000 Диапазон: 1-1000

Plugins.Log.MaxLinesPerSecond

Установите максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок `log`, `logrt` и `eventlog`. Указанное значение будет переопределено параметром `maxlines`, заданным в ключе элемента данных `log`, `logrt` или `eventlog`.
Примечание: Zabbix будет обрабатывать в 10 раз больше новых строк, чем задано в `MaxLinesPerSecond`, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

Plugins.SystemRun.LogRemoteCommands

Включить журналирование выполненных команд оболочки как предупреждений. Команды будут записываться в журнал только при удаленном выполнении. Записи в журнале не будут создаваться, если `system.run[]` запускается локально параметрами `HostMetadataItem`, `HostInterfaceItem` или `HostnameItem`.

По умолчанию: 0
 Значения: 0 - отключено, 1 - включено

PluginSocket

Укажите путь к именованному каналу Windows для взаимодействия загружаемых плагинов.

По умолчанию: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Установите время ожидания для соединений с загружаемыми плагинами в секундах.

По умолчанию: `Timeout`
 Диапазон: 1-30

ProfilerDir

Каталог, в который будут записываться данные профилирования.

Значение по умолчанию: c:\zabbix_pprof

ProfilerMaxFilesPerProfile

Максимальное количество файлов, сохраняемых для каждого типа профиля. Ограничение применяется к завершённым файлам после каждого дампа профиля. Активный профиль ЦП собирается в скрытом временном файле и не учитывается при расчёте ограничения. На файлы, созданные с помощью periodic_prof_execute, также распространяется это ограничение.

Значение по умолчанию: 10
 Диапазон: 1–100

ProfilerInterval

Интервал записи данных профилирования в секундах.

Значение по умолчанию: 3600
 Диапазон: 1–86400

RefreshActiveChecks

Укажите частоту обновления списка активных проверок в секундах.
 Обратите внимание, что после неудачной попытки обновить активные проверки следующая попытка обновления будет выполнена через 60 секунд.

По умолчанию: 5
 Диапазон: 1-86400

Сервер

Укажите IP-адреса, при необходимости в нотации CIDR, или DNS-имена серверов/прокси Zabbix. Список, разделенный запятыми.
 Если параметр не указан, пассивные проверки отключены, и агент не будет прослушивать ни один TCP-порт.
 Входящие соединения будут приниматься только от узлов сети, перечисленных здесь. Если поддержка IPv6 включена, то 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 считаются эквивалентными, и ::/0 позволит использовать любой IPv4- или IPv6-адрес. 0.0.0.0/0 можно использовать, чтобы разрешить любой IPv4-адрес. Пробелы допускаются.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Укажите адрес сервера/прокси Zabbix или конфигурацию кластера, откуда получать активные проверки.
 Если параметр не указан, активные проверки отключены.

Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделенные двоеточием. Если порт не указан, используется порт по умолчанию.
 IPv6-адреса должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для IPv6-адресов необязательны.

Конфигурация кластера — это один или несколько адресов участников группы серверов или прокси, разделенных точкой с запятой.

Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделяя их запятыми.

Если не используются группы прокси, для каждого сервера/кластера Zabbix не следует указывать более одного прокси Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.

Можно указать несколько адресов, разделенных запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Допускаются пробелы.

Пример для прокси Zabbix:

```
ServerActive=127.0.0.1:10051
```

Пример для группы прокси Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Пример для нескольких серверов:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Пример для высокой доступности:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Пример для высокой доступности с двумя кластерами и одним сервером:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Укажите исходящий IP-адрес для:

- исходящих подключений к серверу Zabbix или прокси Zabbix;
- установления соединений при выполнении некоторых элементов данных (`web.page.get`, `net.tcp.port` и т. д.).

StatusPort

Укажите порт, на котором агент прослушивает HTTP-запросы состояния (`http://localhost:<port>/status`).

Диапазон: 1024-32767

Таймаут

Укажите, сколько времени ожидать (в секундах) при установлении соединения и обмене данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций связи, включая:

- ожидание ответа от сервера Zabbix;
- отправку запросов на сервер Zabbix, включая запросы на конфигурацию элементов данных и данные элементов данных в активных проверках;
- получение данных журнала через мониторинг logfile или журнала событий Windows;
- отправку сообщений heartbeat;
- максимальную длительность проверок `vfs.*`;
- использование в качестве резервного варианта в сценариях, когда сервер или прокси старше версии 7.0 отправляет проверки без таймаутов.

Этот таймаут **не** будет использоваться для тех проверок агента, для которых на веб-интерфейсе можно настроить параметры таймаута (на глобальном уровне, уровне прокси или на уровне отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Укажите уровень шифрования, требуемый для входящих подключений.
 Используется для пассивных проверок.

Можно указать несколько значений, разделяя их запятыми:

- *unencrypted* - принимать подключения без шифрования (по умолчанию);
- *psk* - принимать подключения с TLS и предварительно общим ключом (PSK);
- *cert* - принимать подключения с TLS и сертификатом.

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла-пира, используемые для зашифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованного обмена данными с компонентами Zabbix.

TLSCipherAll13

Укажите строку шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.

Пример:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Укажите строку шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.

Обратите внимание, что этот параметр нельзя использовать вместе с `TLSAccept=cert,psk`; для подключения по сертификату (`TLSConnect=cert`) используйте вместо него `TLSCipherAll13`.

TLSCipherPSK13

Укажите строку шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Укажите уровень шифрования, требуемый для подключений агента к серверу Zabbix или прокси.
 Используется для активных проверок.
 Можно указать только одно значение: - *unencrypted* - подключение без шифрования (по умолчанию); - *psk* - подключение с использованием TLS и предварительно общего ключа (PSK); - *cert* - подключение с использованием TLS и сертификата.

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для подключения *unencrypted*); в противном случае нет

TLSCRLFile

Укажите полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных соединений между компонентами Zabbix.

TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованной связи между компонентами Zabbix.

TLSPSKFile

Укажите полный путь к файлу, содержащему предварительно общий ключ агента, используемый для зашифрованной связи с сервером Zabbix или прокси.

TLSPSKIdentity

Укажите строку идентификатора предварительно разделенного ключа, используемую для зашифрованной связи с сервером или прокси Zabbix.

TLSServerCertIssuer

Укажите допустимого издателя сертификата сервера (прокси).

TLSServerCertSubject

Укажите допустимый subject сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передавать все символы в аргументах пользовательских параметров. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не допускаются.

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

UserParameter

Укажите пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: `UserParameter=<key>,<shell command>`
Обратите внимание, что команда оболочки не должна возвращать пустую строку или только EOL. Команды оболочки могут использовать относительные пути, если указан параметр `UserParameterDir`.

Пример:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Укажите путь поиска по умолчанию для команд `UserParameter`. Если этот параметр используется, агент изменит свой рабочий каталог на указанный здесь перед выполнением команды. Таким образом, команды UserParameter могут использовать относительный префикс `.\` вместо полного пути.
Допускается только одна запись.

Пример:

```
UserParameterDir=C:\opt\myscripts
```

Параметры конфигурации: плагины

См. параметры конфигурации для плагинов агента Zabbix 2:

- Плагин Ceph
- Плагин Docker
- Плагин Ember+
- Плагин Memcached
- Плагин Modbus
- Плагин MongoDB
- Плагин MQTT
- Плагин MSSQL
- Плагин MySQL
- Плагин NVIDIA GPU
- Плагин Oracle
- Плагин PostgreSQL
- Плагин Redis
- Плагин S.M.A.R.T.

Принципы настройки плагинов

На этой странице описаны принципы настройки плагинов и рекомендации по их применению.

Все плагины настраиваются с помощью параметров `Plugins.*`, которые могут находиться в файле конфигурации агента 2 Zabbix или в файле конфигурации плагина.
Если плагин использует отдельный файл конфигурации, путь к этому файлу необходимо указать в параметре `Include` файла конфигурации агента 2 Zabbix.

Типичный параметр плагина имеет следующую структуру:

`Plugins.<PluginName>.<Parameter>=<Value>`

Кроме того, существуют две специальные группы параметров:

- `Plugins.<PluginName>.Default.<Parameter>=<Value>` используется для определения значений параметров по умолчанию.
- `Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>` используется для определения отдельных наборов параметров для разных целей мониторинга с помощью именованных сеансов.

Имена всех параметров должны соответствовать следующим требованиям:

- Рекомендуется использовать прописные буквы в именах плагинов.
- Параметр должен начинаться с прописной буквы.
- Специальные символы не допускаются.
- Максимальный уровень вложенности не ограничен.
- Количество параметров не ограничено.

Например, чтобы выполнять активные проверки, у которых нет интервала обновления *Scheduling*, сразу после перезапуска агента только для плагина Uptime, укажите `Plugins.Uptime.System.ForceActiveChecksOnStart=1` в файле конфигурации.

Аналогично, чтобы задать пользовательское ограничение для параллельных проверок плагина CPU, укажите `Plugins.CPU.System.` в файле конфигурации.

Значения по умолчанию

Вы можете задать значения по умолчанию для параметров, связанных с подключением (URI, имя пользователя, пароль и т. д.), в файле конфигурации в следующем формате:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Например: `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017` и т. д.

Если значение такого параметра не указано в ключе элемента данных или параметрах именованного сеанса, плагин будет использовать значение по умолчанию. Если параметр по умолчанию также не определен, будут использоваться жестко заданные значения по умолчанию.

Note:
Если ключ элемента данных не содержит параметров, агент 2 Zabbix попытается собрать метрику, используя значения, заданные в разделе параметров по умолчанию.

Именованные сеансы

Именованные сеансы представляют собой дополнительный уровень параметров плагина и могут использоваться для задания отдельных наборов параметров аутентификации для каждого из контролируемых экземпляров. Каждый параметр именованного сеанса должен иметь следующую структуру:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Имя сеанса можно использовать как параметр ключа элемента данных connString вместо отдельного указания URI, имени пользователя и/или пароля.

В ключах элементов данных первым параметром может быть либо connString, либо URI. Если первый параметр ключа не совпадает ни с одним именем сеанса, он будет обработан как URI. Обратите внимание, что передача встроенных учетных данных URI в ключе элемента данных не поддерживается, вместо этого используйте параметры именованного сеанса.

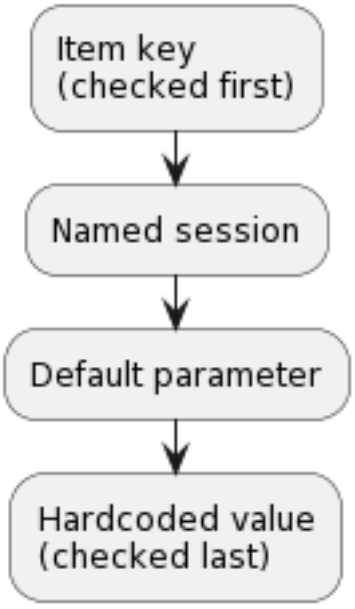
Список доступных параметров именованного сеанса зависит от плагина.

Параметры сеанса можно переопределить, указав новые значения в параметрах ключа элемента данных (см. пример).

Если параметр не определен для именованного сеанса, Zabbix agent 2 будет использовать значение, заданное в параметре плагина по умолчанию.

Приоритет параметров

Плагины Zabbix agent 2 ищут значения параметров, связанных с подключением, в следующем порядке:



1. Первый параметр ключа элемента данных сравнивается с именами сеансов. Если совпадение не найдено, он считается фактическим значением; в этом случае шаг 3 будет пропущен. Если совпадение найдено, значение параметра (обычно URI) должно быть определено в именованном сеансе.
2. Остальные параметры будут взяты из ключа элемента данных, если они определены.
3. Если параметр ключа элемента данных (например, password) пуст, плагин будет искать соответствующий параметр именованного сеанса.
4. Если параметр сеанса также не указан, будет использовано значение, заданное в соответствующем значении по умолчанию.
5. Если ничего из этого не сработает, плагин использует жестко заданное значение по умолчанию.

Пример 1

Мониторинг двух экземпляров "MySQL1" и "MySQL2".

Параметры конфигурации:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
```

```
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

В результате такой конфигурации каждое имя сеанса может использоваться как connString в ключе элемента данных, например `mysql.ping[MySQL1]` или `mysql.ping[MySQL2]`.

Пример 2

Передача части параметров в ключе элемента данных.

Параметры конфигурации:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

Ключ элемента данных: `pgsql.ping[session1,new_username,,postgres]`

В результате такой конфигурации агент подключится к PostgreSQL, используя следующие параметры:

- URI из параметра сеанса: `192.0.2.234:5432`
- Имя пользователя из ключа элемента данных: `new_username`
- Пароль из параметра сеанса (так как он опущен в ключе элемента данных): `session_password`
- Имя базы данных из ключа элемента данных: `postgres`

Пример 3

Сбор метрики с использованием параметров конфигурации по умолчанию.

Параметры конфигурации:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Ключ элемента данных: `pgsql.ping[,,,postgres]`

В результате такой конфигурации агент подключится к PostgreSQL, используя следующие параметры:

- URI по умолчанию: `192.0.2.234:5432`
- Имя пользователя по умолчанию: `zabbix`
- Пароль по умолчанию: `password`
- Имя базы данных из ключа элемента данных: `postgres`

Соединения

Некоторые плагины поддерживают сбор метрик одновременно с нескольких экземпляров. Можно отслеживать как локальные, так и удаленные экземпляры. Поддерживаются TCP-соединения и соединения через Unix-сокет.

Рекомендуется настраивать плагины так, чтобы соединения с экземплярами оставались открытыми. Преимущества этого подхода — снижение сетевой нагрузки, задержки, а также использования CPU и памяти за счет меньшего числа соединений. За это отвечает клиентская библиотека.

> **Note:**
> Период времени, в течение которого неиспользуемые соединения должны оставаться открытыми, можно задать с помощью параметра `Plugins.<PluginName>.KeepAlive`. Пример: `Plugins.Memcached.KeepAlive`

Плагин Ceph

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Ceph.

Рекомендуется указывать их в отдельном файле конфигурации (например, `ceph.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию агента Zabbix 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Ceph.Default.ApiKey	Укажите ключ API по умолчанию для подключения к Ceph.
Plugins.Ceph.Default.Mode	Укажите режим по умолчанию для подключения к Ceph.
Plugins.Ceph.Default.URI	Укажите URI по умолчанию для подключения к Ceph.
Plugins.Ceph.Default.User	Укажите имя пользователя по умолчанию для подключения к Ceph.
Plugins.Ceph.InsecureSkipVerify	Укажите, должен ли HTTP-клиент проверять цепочку сертификатов сервера и имя узла сети.
Plugins.Ceph.KeepAlive	Задайте максимальное время ожидания перед закрытием неиспользуемых подключений плагина.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Укажите ключ API именованной сессии для подключения к Ceph.
Plugins.Ceph.Sessions.<SessionName>.Mode	Укажите режим именованной сессии для подключения к Ceph.
Plugins.Ceph.Sessions.<SessionName>.URI	Укажите URI именованной сессии для подключения к Ceph.
Plugins.Ceph.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии для подключения к Ceph.
Plugins.Ceph.Timeout	Задайте тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Сведения о параметре

Plugins.Ceph.Default.ApiKey

Укажите ключ API по умолчанию для подключения к Ceph.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Ceph.Default.Mode

Укажите режим по умолчанию для подключения к Ceph.
 Нативный режим поддерживается только в Linux и, начиная с Ceph 16. Режим restful не будет работать с версией Ceph 20 или новее.
 Обратите внимание, что набор учетных данных пользователя отличается для каждого режима, и они несовместимы друг с другом.

По умолчанию: `restful` (устарело)
 Диапазон: native/restful

Plugins.Ceph.Default.URI

Укажите URI по умолчанию для подключения к Ceph.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Не должен содержать встроенные учетные данные (они будут проигнорированы).
Должен соответствовать формату URI.
Поддерживается только схема `https`; схему можно опустить.
Порт можно опустить (по умолчанию=8003).

По умолчанию: `https://localhost:8003`

Примеры:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Укажите имя пользователя по умолчанию для подключения к Ceph.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Ceph.InsecureSkipVerify

Укажите, должен ли HTTP-клиент проверять цепочку сертификатов и имя хоста сервера.
 Если *true*, TLS принимает любой сертификат, представленный сервером, и любое имя хоста в этом сертификате. В этом режиме TLS уязвим для атак типа man-in-the-middle (используйте только для тестирования).

По умолчанию: `false`
 Диапазон: false/true

Plugins.Ceph.KeepAlive

Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Укажите API-ключ именованной сессии для подключения к Ceph.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.Ceph.Sessions.<SessionName>.Mode

Укажите именованный режим сеанса для подключения к Ceph.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.
 Нативный режим поддерживается только в Linux и начиная с Ceph 16. Режим restful не будет работать с версией Ceph 20 или новее.
 Обратите внимание, что набор учетных данных пользователя отличается для каждого режима, и они несовместимы друг с другом.

По умолчанию: `restful` (устарело)
 Диапазон: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Укажите URI именованной сессии для подключения к Ceph.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.
Не должен содержать встроенные учетные данные (они будут проигнорированы).
Должен соответствовать формату URI.
 Поддерживается только схема `https`; схему можно опустить.
Порт можно не указывать (по умолчанию=8003).

По умолчанию: `https://localhost:8003`

Примеры:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии, используемой для подключения к Ceph.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.Ceph.Timeout

Установите тайм-аут выполнения запроса (время ожидания завершения запроса перед его принудительным завершением) в секундах.

По умолчанию: `global timeout`
 Диапазон: 1-30

См. также

- Инструкции по настройке плагинов
- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

Плагин Docker

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Docker.

Рекомендуется указывать их в отдельном файле конфигурации (например, `docker.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Docker.Endpoint	Укажите расположение UNIX-сокета демона Docker..
Plugins.Docker.Timeout	Укажите тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;

- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Подробности параметра

Plugins.Docker.Endpoint

Укажите расположение UNIX-сокета демона Docker.
 Должно содержать схему (поддерживается только `unix://`).

По умолчанию: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Укажите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах перед его принудительным завершением).

По умолчанию: `global timeout`
 Диапазон: 1–30

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин Ember+

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Ember+.

Рекомендуется указывать их в отдельном файле конфигурации (например, `emberplus.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Плагин Ember+ является загружаемым плагином и доступен, а также полностью описан в репозитории плагина Ember+.

В настоящее время этот плагин доступен только для сборки из исходного кода (как для Unix, так и для Windows).

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница структурирована следующим образом:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.EmberPlus.Default.Uri	Укажите URI по умолчанию для подключения к Ember+.
Plugins.EmberPlus.KeepAlive	Установите максимальное время ожидания, после которого неиспользуемые подключения плагина будут закрыты.
Plugins.EmberPlus.SessionsX.SessionName	Укажите имя именованной сессии для подключения к Ember+.
Plugins.EmberPlus.SessionsX.Uri	Укажите URI именованной сессии для подключения к Ember+.
Plugins.EmberPlus.System.Path	Укажите путь к исполняемому файлу плагина Ember+.
Plugins.EmberPlus.Timeout	Установите тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Подробности параметра

Plugins.EmberPlus.Default.Uri

Укажите URI по умолчанию для подключения к Ember+.
 Единственная поддерживаемая схема — `tcp://`. Схему можно опустить. Встроенные учетные данные будут проигнорированы.

По умолчанию: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60–900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к Ember+.
 Поддерживается только схема `tcp://`. Схему можно опустить. Встроенные учетные данные будут проигнорированы.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

По умолчанию: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Укажите путь к исполняемому файлу плагина Ember+.

Пример: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Установите длительность ожидания ответа сервера в секундах при первом подключении и при последующих операциях в сеансе.

По умолчанию: `global timeout`
 Диапазон: 1-30

Параметры

Parameter	Description
-V --version	Вывести информацию о версии плагина и лицензии.
-h --help	Вывести справочную информацию (сокращённый вариант).
-t, --test <item key>	Запустить плагин для тестирования (конфигурация плагина игнорируется).

См. также:

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин Memcached

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Memcached.

Рекомендуется указывать их в отдельном файле конфигурации (например, `memcached.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Memcached.DefaultPassword	Укажите пароль по умолчанию для подключения к Memcached.
Plugins.Memcached.DefaultURI	Укажите URI по умолчанию для подключения к Memcached.
Plugins.Memcached.DefaultUsername	Укажите имя пользователя по умолчанию для подключения к Memcached.
Plugins.Memcached.KeepAlive	Задайте максимальное время ожидания перед закрытием неиспользуемых подключений плагина.
Plugins.Memcached.Sessions.<SessionName>.Password	Укажите пароль именованной сессии для подключения к Memcached.
Plugins.Memcached.Sessions.<SessionName>.URI	Укажите URI именованной сессии для подключения к Memcached.
Plugins.Memcached.Sessions.<SessionName>.Username	Укажите имя пользователя именованной сессии для подключения к Memcached.
Plugins.Memcached.Timeout	Задайте тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;

- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Plugins.Memcached.Default.Password

Укажите пароль по умолчанию для подключения к Memcached.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Memcached.Default.Uri

Укажите URI по умолчанию для подключения к Memcached.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI. Поддерживаемые схемы: `tcp`, `unix`.
 Схему можно опустить.
 Порт можно опустить (по умолчанию=11211).

По умолчанию: `tcp://localhost:11211`

Примеры:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Укажите имя пользователя по умолчанию для подключения к Memcached.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Memcached.KeepAlive

Задайте максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60-900

Plugins.Memcached.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии, используемой для подключения к Memcached.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.Memcached.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к Memcached.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI. Поддерживаемые схемы: `tcp`, `unix`.
 Схему можно опустить.
 Порт можно опустить (по умолчанию=11211).

Примеры:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии при подключении к Memcached.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.Memcached.Timeout

Установите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах, после которой он будет принудительно завершен).

По умолчанию: `global timeout`
 Диапазон: 1-30

См. также:

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин Modbus

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Modbus.

Рекомендуется указывать их в отдельном файле конфигурации (например, `modbus.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница структурирована следующим образом:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Укажите именованную конечную точку сеанса с именем для подключения к Modbus.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Укажите идентификатор Slave для сеанса с именем для подключения к Modbus.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Установите тайм-аут сеанса с именем в секундах.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Сведения о параметре

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Укажите именованную конечную точку сеанса для подключения к Modbus.
 Строка подключения состоит из схемы протокола, адреса узла сети и порта или имени последовательного порта, а также атрибутов.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Укажите Slave ID именованной сессии для подключения к Modbus.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Пример: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Примечание: этот параметр именованной сессии проверяется только если значение, указанное в параметре slave ID ключа элемента данных, пустое.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Установите тайм-аут указанной сессии в секундах.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Пример: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Если вам нужно задать тайм-аут выполнения запроса (сколько ждать завершения запроса перед его остановкой), используйте форму настройки элемента данных.

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин MongoDB

Обзор

Параметры конфигурации позволяют настраивать поведение плагина MongoDB.

Рекомендуется указывать их в отдельном файле конфигурации (например, `mongodb.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Плагин MongoDB является загружаемым плагином и доступен, а также полностью описан в репозитории плагина MongoDB.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

При подключении к MongoDB плагин проверяет свою конфигурацию в определенном порядке, который определяет, какое значение использовать. Порядок, от наивысшего к наинизшему приоритету, следующий:

1. Параметры, указанные в URI подключения (например, параметры запроса, заданные после ? в `Plugins.MongoDB.Sessions.<` имеют наивысший приоритет).
2. Параметры в ключе элемента данных Zabbix agent 2 (например, mongodb.collection.stats).
3. Параметры, определенные в именованной сессии (например, `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions.<SessionName>.Password`).
4. Значения по умолчанию (например, `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Например:

- Если вы укажете базу данных для аутентификации подключения к серверу MongoDB в параметре `*.<SessionName>.Uri` (например, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSou` то плагин будет считать, что параметр `*.<SessionName>.AuthSource` установлен в `admin`, даже если параметр `*.<SessionName>.AuthSource` задан другим значением.
- Если вы укажете URI в параметре `*.<SessionName>.Uri` (например, `Plugins.MongoDB.Sessions.<session_name>.Uri=mo` но укажете имя пользователя и пароль в ключе элемента данных Zabbix agent 2, плагин будет использовать имя пользователя и пароль из ключа элемента данных, даже если настроены `Plugins.MongoDB.Sessions.<SessionName>.User` и `Plugins.MongoDB.Sessions.<SessionName>.Password`.

Attention:
Если используется схема `mongodb+srv://`, URI может получать параметры из DNS TXT-записи. Параметры, полученные из этой TXT-записи, переопределяют все остальные настройки, включая параметры URI, параметры ключа элемента данных Zabbix Agent 2 и параметры сессии.
 Для имени хоста, используемого в строке подключения, может существовать только одна TXT-запись.

Страница состоит из следующих разделов:

- Обзор параметров (чтобы увидеть полные сведения о параметре, щелкните его имя)
- Сведения о параметрах

Parameter	Description
Plugins.MongoDB.Default.AuthMechanism	Укажите механизм аутентификации по умолчанию для подключения к серверу MongoDB.
Plugins.MongoDB.Default.AuthSource	Укажите базу данных по умолчанию для аутентификации подключения к серверу MongoDB.
Plugins.MongoDB.Default.Direct	Отключить автоматическое обнаружение узлов сети реплика-сета MongoDB.
Plugins.MongoDB.Default.Password	Укажите пароль по умолчанию.
Plugins.MongoDB.Default.ReplicaSet	Укажите имя реплика-сета по умолчанию, используемое для автоматического обнаружения узлов сети реплика-сета MongoDB.
Plugins.MongoDB.Default.ReadPreference	Укажите порядок по умолчанию для подключения к узлам сети реплика-сета MongoDB или резервного перехода.
Plugins.MongoDB.Default.TLSConnect	Укажите уровень шифрования, требуемый для обмена данными между Zabbix agent 2 и контролируемыми базами данных.
Plugins.MongoDB.Default.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла.
Plugins.MongoDB.Default.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов.
Plugins.MongoDB.Default.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ базы данных.
Plugins.MongoDB.Default.Uri	Укажите URI по умолчанию.
Plugins.MongoDB.Default.User	Укажите имя пользователя по умолчанию.
Plugins.MongoDB.KeepAlive	Установите максимальное время ожидания перед закрытием неиспользуемых подключений плагина.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	Укажите механизм аутентификации именованной сессии для подключения к серверу MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Укажите базу данных именованной сессии для аутентификации подключения к серверу MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.Direct	Отключить автоматическое обнаружение узлов сети реплика-сета MongoDB для именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.Password	Укажите пароль именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Укажите имя реплика-сета именованной сессии, используемое для автоматического обнаружения узлов сети реплика-сета MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Укажите порядок именованной сессии для подключения к узлам сети реплика-сета MongoDB или резервного перехода.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Укажите уровень шифрования, требуемый для обмена данными между Zabbix agent 2 и контролируемыми базами данных для именованной сессии.

Parameter	Description
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла для именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов для именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ базы данных для именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Укажите тайм именованной сессии.
Plugins.MongoDB.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии.
Plugins.MongoDB.System.Path	Укажите путь к исполняемому файлу плагина.
Plugins.MongoDB.Timeout	Установите тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные среды.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Plugins.MongoDB.Default.AuthMechanism

Укажите механизм аутентификации по умолчанию для подключения к серверу MongoDB.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

Диапазон: `SCRAM-SHA-1`, `SCRAM-SHA-256` или `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Укажите базу данных по умолчанию для аутентификации при подключении к серверу MongoDB.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

По умолчанию: `admin`

Если используется механизм аутентификации `MONGODB-X509`, этот параметр автоматически устанавливается в `$external`; изменение его на любое другое значение приведет к ошибке при запуске.

Plugins.MongoDB.Default.Direct

Отключает автоматическое обнаружение узлов реплика-сета MongoDB.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

По умолчанию: `false`
 Диапазон: `true` - подключаться только к указанному серверу, игнорируя других участников реплика-сета, `false` - автоматически обнаруживать все узлы в реплика-сете.

Если используется схема `tcp://` (устаревшая), параметр по умолчанию устанавливается в `true`.

Plugins.MongoDB.Default.Password

Укажите пароль по умолчанию.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Plugins.MongoDB.Default.ReplicaSet

Укажите имя набора реплик по умолчанию, используемое для автоматического обнаружения узлов сети набора реплик MongoDB.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.
 Если этот параметр используется со схемой `mongodb+srv://`, он будет работать так, как если бы было указано несколько URL; DNS-сервер может иметь TXT-запись, которая может переопределить этот (или любой другой) параметр.
 Недоступно, если параметр `Plugins.MongoDB.Sessions.<SessionName>.Direct` установлен в `true`; агент не сможет запуститься.

Plugins.MongoDB.Default.ReadPreference

Укажите порядок по умолчанию для подключения к узлам сети репликационного набора MongoDB или резервный вариант.
 <SessionName> - укажите имя сеанса для использования в ключах элементов данных.
 Недоступно, если для параметра `Plugins.MongoDB.Sessions.<SessionName>.Direct` установлено значение `true`; агент не запустится.

Диапазон: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` или `NearestMode`

Рекомендуется использовать значения primary, primaryPreferred, secondary, secondaryPreferred и nearest. Значения, оканчивающиеся на Mode, поддерживаются как устаревшие псевдонимы для обеспечения совместимости.

Plugins.MongoDB.Default.TLSConnect

Укажите уровень шифрования, требуемый для обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.
 Поддерживается начиная с версии плагина 1.2.1.

Диапазон: required - требуется TLS-соединение, verify_ca - проверка сертификатов, verify_full - проверка сертификатов и IP-адреса

Plugins.MongoDB.Default.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата узла, используемые для зашифрованных соединений между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

Обязательно: yes, если Plugins.MongoDB.Sessions.<SessionName>.TLSConnect имеет значение verify_ca или verify_full

Plugins.MongoDB.Default.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованной связи между Zabbix agent 2 и контролируемыми базами данных.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Обязательно: yes, если указан Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Plugins.MongoDB.Default.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ базы данных, используемый для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

Обязательно: yes, если указан Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Plugins.MongoDB.Default.Uri

Укажите URI по умолчанию.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: mongodb:// и mongodb+srv://; схема tcp:// устарела и сохранена для обратной совместимости с существующими конфигурациями.
 Порт можно не указывать (по умолчанию=27017).
Примеры: “

По умолчанию: tcp://127.0.0.1:27017

Примеры:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Укажите имя пользователя по умолчанию.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Plugins.MongoDB.KeepAlive

Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60-900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Укажите механизм аутентификации для именованной сессии при подключении к серверу MongoDB.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Диапазон: SCRAM-SHA-1, SCRAM-SHA-256 или MONGODB-X509

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Укажите базу данных именованной сессии для аутентификации подключения к серверу MongoDB.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

По умолчанию: admin

Если используется механизм аутентификации MONGODB-X509, этот параметр автоматически устанавливается в $external; изменение его на любое другое значение приведет к ошибке при запуске.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Отключает автоматическое обнаружение узлов репликационного набора MongoDB для именованной сессии.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

По умолчанию: false
 Диапазон: true - подключаться только к указанному серверу, игнорируя других участников репликационного набора, false - автоматически обнаруживать все узлы в репликационном наборе.

Если используется схема tcp:// (устарела), параметр по умолчанию устанавливается в true.

Plugins.MongoDB.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Укажите имя набора реплик для именованной сессии, используемое для автоматического обнаружения узлов сети в наборе реплик MongoDB.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Если этот параметр используется со схемой mongodb+srv://, он будет работать так, как если бы было указано несколько URL; DNS-сервер может иметь TXT-запись, которая может переопределить этот (или любой другой) параметр.
 Недоступно, если параметр Plugins.MongoDB.Sessions.<SessionName>.Direct установлен в true; агент не сможет запуститься.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Укажите порядок сеансов с именами для подключения к узлам репликационного набора MongoDB или резервного подключения.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.
 Недоступно, если для параметра Plugins.MongoDB.Sessions.<SessionName>.Direct задано значение true; агент не запустится.

Диапазон значений: primary, primaryPreferred, secondary, secondaryPreferred, nearest, PrimaryMode, PrimaryPreferredMode, SecondaryMode, SecondaryPreferredMode или NearestMode

Рекомендуется использовать значения primary, primaryPreferred, secondary, secondaryPreferred и nearest. Значения, оканчивающиеся на Mode, поддерживаются как устаревшие псевдонимы для обеспечения совместимости.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Укажите уровень шифрования, требуемый для обмена данными между агентом Zabbix agent 2 и контролируемыми базами данных для именованной сессии.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.
 Поддерживается начиная с версии плагина 1.2.1.

Диапазон: required - требуется TLS-соединение, verify_ca - проверка сертификатов, verify_full - проверка сертификатов и IP-адреса

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла, используемые для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Обязательно: yes, если Plugins.MongoDB.Sessions.<SessionName>.TLSConnect установлено в verify_ca или verify_full

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованной связи между Zabbix agent 2 и контролируемыми базами данных для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Обязательно: yes, если указан Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ базы данных, используемый для зашифрованной связи между агентом 2 Zabbix и контролируемыми базами данных для именованного сеанса.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Обязательно: yes, если указан Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Plugins.MongoDB.Sessions.<SessionName>.Uri

Укажите URI именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: `mongodb://` и `mongodb+srv://`; схема `tcp://` устарела и сохранена для обратной совместимости с существующими конфигурациями.
 Порт можно не указывать (по умолчанию=27017).
Примеры: “

По умолчанию: `tcp://127.0.0.1:27017`

Примеры:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MongoDB.System.Path

Укажите путь к исполняемому файлу плагина MongoDB.

Пример использования: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb`

Plugins.MongoDB.Timeout

Установите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах, после которой он будет принудительно остановлен).

По умолчанию: `global timeout`
 Диапазон: 1-30

Параметры

Parameter	Description
-V --version	Вывести версию плагина и информацию о лицензии.
-h --help	Вывести справочную информацию (сокращенный вариант).
-t, --test <item key>	Запустить плагин для тестирования (конфигурация плагина игнорируется).

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин MQTT

Обзор

Параметры конфигурации позволяют настраивать поведение плагина MQTT.

Рекомендуется указывать их в отдельном файле конфигурации (например, `mqtt.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.MQTT.Default.Password	Укажите пароль по умолчанию для подключения к MQTT.
Plugins.MQTT.Default.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла.
Plugins.MQTT.Default.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов.
Plugins.MQTT.Default.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ MQTT.
Plugins.MQTT.Default.Topic	Укажите тему по умолчанию для подписки MQTT.
Plugins.MQTT.Default.Url	Укажите строку подключения к брокеру MQTT по умолчанию.

Parameter	Description
Plugins.MQTT.Default.User	Укажите имя пользователя по умолчанию для подключения к MQTT.
Plugins.MQTT.Sessions.<SessionName>.Password	Укажите пароль именованной сессии для подключения к MQTT.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла для именованной сессии.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов для именованной сессии.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ MQTT для именованной сессии.
Plugins.MQTT.Sessions.<SessionName>.Topic	Укажите тему именованной сессии для подписки MQTT.
Plugins.MQTT.Sessions.<SessionName>.Url	Укажите строку подключения к брокеру MQTT для именованной сессии.
Plugins.MQTT.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии для подключения к MQTT.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Сведения о параметре

Plugins.MQTT.Default.Password

Укажите пароль по умолчанию для подключения к MQTT.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.MQTT.Default.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла при зашифрованном обмене данными между агентом Zabbix 2 и MQTT-брокером.
 Используется, если в именованной сессии значение не указано.

Plugins.MQTT.Default.TLSCertFile

Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов для зашифрованной связи между агентом Zabbix 2 и MQTT broker.
 Используется, если в именованной сессии не указано значение.

Plugins.MQTT.Default.TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ MQTT для зашифрованной связи между агентом Zabbix 2 и MQTT-брокером.
 Используется, если в именованной сессии не указано значение.

Plugins.MQTT.Default.Topic

Укажите тему по умолчанию для подписки MQTT.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.
 Тема может содержать подстановочные знаки (”+”,”#”).

Примеры:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Укажите строку подключения по умолчанию для MQTT broker.
 Используется, если в ключе элемента данных или именованной сессии не указано значение.
 Не должно включать параметры запроса.
 Должно соответствовать формату URL.
 Поддерживаемые схемы: `tcp` (по умолчанию), `ws`, `tls`
 Схему можно опустить.
 Порт можно опустить (по умолчанию=1883)

По умолчанию: `tcp://localhost:1883`

Примеры:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Укажите имя пользователя по умолчанию для подключения к MQTT.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.MQTT.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии для подключения к MQTT.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Укажите полное имя пути к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла для указанной сессии, используемой для зашифрованного обмена данными между агентом Zabbix 2 и MQTT broker.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов для именованной сессии, используемой для зашифрованной связи между Zabbix агент 2 и MQTT broker.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ MQTT для именованной сессии, используемый для зашифрованной связи между агентом Zabbix 2 и брокером MQTT.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MQTT.Sessions.<SessionName>.Topic

Укажите тему именованной сессии для подписки MQTT.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.
 Тема может содержать подстановочные знаки (”+”,”#”).

Примеры:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

Укажите строку подключения к MQTT-брокеру для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Не должно включать параметры запроса.
 Должно соответствовать формату URL.
 Поддерживаемые схемы: `tcp` (по умолчанию), `ws`, `tls`
 Схему можно не указывать.
 Порт можно не указывать (по умолчанию=1883)

По умолчанию: `tcp://localhost:1883`

Примеры:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии, используемой для подключения к MQTT.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Если вам нужно задать тайм-аут выполнения запроса (как долго ждать завершения запроса перед его остановкой), используйте форму настройки элемента данных.

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин MSSQL

Обзор

Параметры конфигурации позволяют настраивать поведение плагина MSSQL.

Рекомендуется указывать их в отдельном файле конфигурации (например, `mssql.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию агента Zabbix 2.

Плагин MSSQL является загружаемым плагином и доступен, а также полностью описан в репозитории плагина MSSQL.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.MSSQL.CustomQueriesDir	Укажите путь к каталогу, содержащему пользовательские файлы `.sql` с пользовательскими запросами, которые плагин может выполнять.
Plugins.MSSQL.CustomQueriesEnabled	Включите выполнение ключа элемента данных `mssql.custom.query`.
Plugins.MSSQL.Default.CAPath	Укажите путь к файлу сертификата открытого ключа центра сертификации (CA), выдавшего сертификат сервера MSSQL, по умолчанию.
Plugins.MSSQL.Default.Database	Укажите имя базы данных по умолчанию.
Plugins.MSSQL.Default.Encrypt	Укажите уровень шифрования по умолчанию.
Plugins.MSSQL.Default.HostnameInCertificate	Укажите общее имя (CN) сертификата сервера MSSQL по умолчанию.
Plugins.MSSQL.Default.Password	Укажите пароль по умолчанию.
Plugins.MSSQL.Default.TLSMinVersion	Укажите минимальную версию TLS по умолчанию.
Plugins.MSSQL.Default.TrustServerCertificate	Доверять сертификату сервера без проверки его подлинности по умолчанию.
Plugins.MSSQL.Default.Uri	Укажите URI по умолчанию.
Plugins.MSSQL.Default.User	Укажите имя пользователя по умолчанию.
Plugins.MSSQL.KeepAlive	Укажите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.
Plugins.MSSQL.Sessions.<SessionName>.CAPath	Укажите путь к файлу сертификата открытого ключа центра сертификации (CA), выдавшего сертификат сервера MSSQL, для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.Database	Укажите имя базы данных для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Укажите уровень шифрования для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.HostnameInCertificate	Укажите общее имя (CN) сертификата сервера MSSQL для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.Password	Укажите пароль для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Укажите минимальную версию TLS для именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Доверять сертификату сервера именованной сессии без проверки его подлинности.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Укажите URI именованной сессии.
Plugins.MSSQL.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии.
Plugins.MSSQL.System.Path	Укажите путь к исполняемому файлу плагина MSSQL.
Plugins.MSSQL.Timeout	Установите время ожидания в секундах ответа сервера при первом подключении и при последующих операциях в сессии.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Plugins.MSSQL.CustomQueriesDir

Укажите путь к каталогу, содержащему пользовательские файлы `.sql` с пользовательскими запросами, которые может выполнять плагин.
 Плагин загружает все доступные файлы .sql в настроенном каталоге при запуске. Это означает, что любые изменения в файлах пользовательских запросов не будут применены, пока плагин не будет перезапущен. Плагин запускается и останавливается вместе с агентом Zabbix 2.

По умолчанию:
`/usr/local/share/zabbix/custom-queries/mssql` - для систем Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` - для систем Windows (где `*` - имя диска из переменной среды `ProgramFiles`)

Plugins.MSSQL.CustomQueriesEnabled

Включает выполнение ключа элемента данных `mssql.custom.query`.
 Если отключено, запросы не будут загружаться из пути к каталогу пользовательских запросов.

По умолчанию: `false`
 Диапазон: `true/false`

Plugins.MSSQL.Default.CACertPath

Укажите путь к файлу по умолчанию для сертификата открытого ключа центра сертификации (CA), выдавшего сертификат MSSQL-сервера.
 Сертификат должен быть в формате PEM.

Plugins.MSSQL.Default.Database

Укажите имя базы данных по умолчанию.

Plugins.MSSQL.Default.Encrypt

Укажите уровень шифрования по умолчанию.

Диапазон:
true - передача данных между плагином и сервером шифруется;
false - передача данных между плагином и сервером не шифруется, кроме пакета входа в систему;
strict - передача данных между плагином и сервером шифруется сквозным образом с использованием TDS8;
disable - передача данных между плагином и сервером не шифруется.

Plugins.MSSQL.Default.HostNameInCertificate

По умолчанию укажите общее имя (CN) сертификата сервера MSSQL.

Plugins.MSSQL.Default.Password

Укажите пароль по умолчанию для защищенного сервера MSSQL.

Plugins.MSSQL.Default.TLSMinVersion

Укажите минимальную версию TLS по умолчанию.

Диапазон: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Доверять сертификату сервера по умолчанию без его проверки.

Диапазон: `true`, `false`

Plugins.MSSQL.Default.Uri

Укажите URI по умолчанию для подключения к серверу MSSQL.
 Встроенные учетные данные будут проигнорированы.
 Единственная поддерживаемая схема — `sqlserver://`.
 Схему можно опустить.

По умолчанию: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Укажите имя пользователя по умолчанию для защищенного сервера MSSQL.

Plugins.MSSQL.KeepAlive

Укажите максимальное время ожидания (в секундах) до закрытия неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60–900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Укажите путь к файлу именованной сессии для публичного ключа сертификата центра сертификации (CA), который выдал сертификат сервера MSSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Сертификат должен быть в формате PEM.

Plugins.MSSQL.Sessions.<SessionName>.Database

Укажите имя базы данных для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Укажите уровень шифрования именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Диапазон:
true - передача данных между плагином и сервером зашифрована;
false - передача данных между плагином и сервером не шифруется, кроме пакета входа в систему;
strict - передача данных между плагином и сервером зашифрована end-to-end с использованием TDS8;
disable - передача данных между плагином и сервером не шифруется.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Укажите общее имя (CN) сертификата сервера MSSQL для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MSSQL.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии защищенного сервера MSSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Укажите минимальную версию TLS для именованной сессии.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Диапазон: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Доверять сертификату сервера указанной сессии без его проверки.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Диапазон: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к серверу MSSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы.
 Единственная поддерживаемая схема — `sqlserver://`.
 Схему можно опустить.

По умолчанию: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии на защищенном сервере MSSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.MSSQL.System.Path

Укажите путь к исполняемому файлу плагина MSSQL.
 Глобальная настройка для плагина MSSQL. Применяется ко всем подключениям.

Пример: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Установите длительность ожидания ответа сервера в секундах при первом подключении и при последующих операциях в сеансе.

По умолчанию: `global timeout`
 Диапазон: 1-30

Параметры

Parameter	Description
-V --version	Вывести версию плагина и информацию о лицензии.
-h --help	Вывести справочную информацию (сокращенный вариант).
-t, --test <item key>	Запустить плагин для тестирования (конфигурация плагина игнорируется).

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин MySQL

Обзор

Параметры конфигурации позволяют настраивать поведение плагина MySQL.

Рекомендуется указывать их в отдельном файле конфигурации (например, `mysql.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Mysql.CallTimeout	Укажите максимальное время ожидания выполнения запроса в секундах.
Plugins.Mysql.CustomQueriesPath	Укажите полный путь к каталогу, содержащему файлы `.sql` с пользовательскими запросами.
Plugins.Mysql.CustomQueriesEnabled	Включите выполнение ключа элемента данных `mysql.custom.query`.
Plugins.Mysql.Default.Password	Укажите пароль по умолчанию.
Plugins.Mysql.Default.TLSConnect	Укажите уровень шифрования по умолчанию для обмена данными между Zabbix agent 2 и контролируемыми базами данных.
Plugins.Mysql.Default.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла.
Plugins.Mysql.Default.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов.
Plugins.Mysql.Default.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ базы данных.
Plugins.Mysql.Default.Uri	Укажите URI по умолчанию.
Plugins.Mysql.Default.User	Укажите имя пользователя по умолчанию.
Plugins.Mysql.KeepAlive	Укажите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.
Plugins.Mysql.Sessions.<SessionName>.Password	Укажите пароль именованной сессии.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Укажите уровень шифрования именованной сессии для обмена данными между Zabbix agent 2 и контролируемыми базами данных.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла именованной сессии.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента именованной сессии или цепочку сертификатов.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ базы данных именованной сессии.
Plugins.Mysql.Sessions.<SessionName>.Uri	Укажите URI именованной сессии.
Plugins.Mysql.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии.
Plugins.Mysql.Timeout	Установите максимальное время ожидания в секундах при необходимости установления соединения.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Plugins.Mysql.CallTimeout

Укажите максимальное время ожидания выполнения запроса в секундах.

По умолчанию: `global timeout`
 Диапазон: 1-30

Plugins.Mysql.CustomQueriesPath

Укажите полный путь к каталогу, содержащему файлы `.sql` с пользовательскими запросами.

По умолчанию:
`/usr/local/share/zabbix/custom-queries/mysql` - для систем Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - для систем Windows (где `*` - имя диска из переменной среды `ProgramFiles`)

Plugins.Mysql.CustomQueriesEnabled

Включить выполнение ключа элемента данных `mysql.custom.query`.
 Если отключено, запросы не будут загружаться из пути к каталогу пользовательских запросов.

По умолчанию: `false`
 Диапазон: `true`/`false`

Plugins.Mysql.Default.Password

Укажите пароль по умолчанию для защищенного сервера MySQL.

Plugins.Mysql.Default.TLSConnect

Укажите уровень шифрования по умолчанию для обмена данными между агентом Zabbix 2 и контролируемыми базами данных.

Диапазон:
required - требовать TLS-соединение;
verify_ca - проверять сертификаты;
verify_full - проверять сертификаты и IP-адрес

Plugins.Mysql.Default.TLSCAFile

Укажите полное имя пути к файлу, содержащему сертификаты корневого CA(ов) для проверки сертификата узла, используемые для зашифрованного обмена данными между агентом 2 Zabbix и контролируемыми базами данных.

Обязательно: `yes`, если `Plugins.Mysql.Default.TLSConnect` установлено в `verify_ca` или `verify_full`

Plugins.Mysql.Default.TLSCertFile

Укажите полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для зашифрованной связи между агентом Zabbix 2 и контролируемыми базами данных.

Обязательно: `yes`, если указан `Plugins.Mysql.Default.TLSKeyFile`

Plugins.Mysql.Default.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ базы данных, используемый для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.

Обязательно: `yes`, если указан `Plugins.Mysql.Default.TLSCertFile`

Plugins.Mysql.Default.Uri

Укажите URI по умолчанию для подключения к серверу MySQL.
 Встроенные учетные данные будут проигнорированы.
 Поддерживаемые схемы: `tcp`, `unix`
 Схему можно опустить.
 Порт можно опустить (по умолчанию=3306).

Примеры:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

Укажите имя пользователя по умолчанию для защищенного сервера MySQL.

Plugins.Mysql.KeepAlive

Укажите максимальное время ожидания (в секундах) до закрытия неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60-900

Plugins.Mysql.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии на защищенном сервере MySQL.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Укажите уровень шифрования для именованной сессии при обмене данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Диапазон:
required - требуется TLS-соединение;
verify_ca - проверка сертификатов;
verify_full - проверка сертификатов и IP-адреса

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата узла-пира в именованной сессии, используемой для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Обязательно: `yes`, если `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` установлен в `verify_ca` или `verify_full`

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента или цепочку сертификатов для указанной сессии, используемые для шифрованной связи между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Обязательно: `yes`, если указан `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile`

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ базы данных для указанной сессии, используемый для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Обязательно: `yes`, если указан `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile`

Plugins.Mysql.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к серверу MySQL.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы.
 Поддерживаемые схемы: `tcp`, `unix`
 Схема может быть опущена.
 Порт может быть опущен (по умолчанию=3306).

Примеры:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии на защищенном сервере MySQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.Mysql.Timeout

Установите максимальное время ожидания в секундах при установлении соединения.

По умолчанию: `global timeout`
 Диапазон: 1–30

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин NVIDIA GPU

Обзор

Параметры конфигурации позволяют настраивать поведение плагина NVIDIA GPU.

Рекомендуется указывать их в отдельном файле конфигурации (например, `nvidiagpu.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Плагин NVIDIA GPU является загружаемым плагином и доступен, а также полностью описан в репозитории плагина NVIDIA GPU.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.NVIDIA.System.Path	Укажите путь к исполняемому файлу плагина NVIDIA GPU.
Plugins.NVIDIA.Timeout	Укажите максимальное время ожидания ответа сервера (в секундах) при попытках подключения и последующих операциях в сеансе.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Сведения о параметре

Plugins.NVIDIA.System.Path

Укажите путь к исполняемому файлу плагина NVIDIA GPU.

Пример:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Укажите максимальное время ожидания ответа сервера (в секундах) при попытках подключения и последующих операциях в сеансе.
 Глобальный таймаут для элемента данных или индивидуальный таймаут элемента данных переопределит это значение, если он больше.

По умолчанию: `global timeout`
 Диапазон: 1-30

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин Oracle

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Oracle.

Рекомендуется указывать их в отдельном файле конфигурации (например, `oracle.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Oracle.CallTimeout	Максимальное время ожидания в секундах для завершения запроса. В настоящее время этот параметр не влияет на работу; вместо него используется настройка item timeout.
Plugins.Oracle.ConnectTimeout	Укажите максимальное время ожидания в секундах для установления соединения.
Plugins.Oracle.CustomQueriesPath	Укажите полный путь к каталогу, содержащему файлы `.sql` с пользовательскими запросами.
Plugins.Oracle.CustomQueriesEnabled	Включите выполнение ключа элемента данных `oracle.custom.query`.
Plugins.Oracle.Default.Password	Укажите пароль по умолчанию.
Plugins.Oracle.Default.Service	Укажите имя службы по умолчанию.
Plugins.Oracle.Default.URI	Укажите URI по умолчанию.
Plugins.Oracle.Default.User	Укажите имя пользователя по умолчанию.
Plugins.Oracle.KeepAlive	Задайте максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.
Plugins.Oracle.ResolveTNS	Укажите, как интерпретировать строку подключения (ConnString) для подключения к серверу Oracle.
Plugins.Oracle.Sessions.<SessionName>.Password	Укажите пароль именованной сессии.
Plugins.Oracle.Sessions.<SessionName>.Service	Укажите имя службы именованной сессии.
Plugins.Oracle.Sessions.<SessionName>.URI	Укажите URI именованной сессии.
Plugins.Oracle.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные среды.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Подробности параметра

Plugins.Oracle.CallTimeout

Максимальное время ожидания в секундах для завершения запроса.
 В настоящее время этот параметр не влияет ни на что. Максимальное время ожидания определяется тайм-аутом элемента данных.

По умолчанию: Timeout[1]
 Диапазон: 1-30

Plugins.Oracle.ConnectTimeout

Укажите максимальное время ожидания в секундах для установления соединения.

По умолчанию: Timeout[1]
 Диапазон: 1-30

Плагины.Oracle.CustomQueriesPath

Укажите полный путь к каталогу, содержащему файлы .sql с пользовательскими запросами.

По умолчанию:
/usr/local/share/zabbix/custom-queries/oracle - для систем Unix,
*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle - для систем Windows (где * - это имя диска из переменной среды ProgramFiles)

Пример:

```
/etc/zabbix/oracle/sql
```

Плагины.Oracle.CustomQueriesEnabled

Включает выполнение ключа элемента данных oracle.custom.query.
 Если отключено, запросы не будут загружаться из пути к каталогу пользовательских запросов.

По умолчанию: false
 Диапазон: true/false

Плагины.Oracle.Default.Password

Укажите пароль по умолчанию для подключения к Oracle.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Плагины.Oracle.Default.Service

Укажите имя службы по умолчанию для подключения к Oracle (SID не поддерживается).
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Oracle.Default.Uri

Укажите URI по умолчанию для подключения к Oracle.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: tcp, tcps
 Схема может быть опущена.
 Порт может быть опущен (по умолчанию=1521).
 Также можно указать ключ TNS или значение TNS в качестве строки подключения. Значение TNS должно быть составлено без пробелов.

Примеры:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Плагины.Oracle.Default.User

Укажите имя пользователя по умолчанию для подключения к Oracle.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Oracle.KeepAlive

Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60-900

Plugins.Oracle.ResolveTNS

Укажите, как интерпретировать строку подключения (ConnString) для подключения к серверу Oracle.
 Если для нее задано значение false, к ConnString будут добавлены схема и порт по умолчанию (если они отсутствуют).
 Если для параметра задано значение true, схема и порт по умолчанию не будут добавлены (если они не указаны явно в ConnString), а ConnString будет передана клиенту Oracle как есть. Если клиент Oracle найдет ConnString в файле tnsnames.ora, для подключения к серверу Oracle будет использовано найденное описание подключения.

По умолчанию: true
 Диапазон: true/false

Plugins.Oracle.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии, используемой для подключения к Oracle.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.Oracle.Sessions.<SessionName>.Service

Укажите имя службы именованной сессии для подключения к Oracle (SID не поддерживается).
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.Oracle.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к Oracle.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: `tcp`, `tcps`
 Схема может быть опущена.
 Порт может быть опущен (по умолчанию=1521).
 Также можно указать ключ TNS или значение TNS в качестве строки подключения. Значение TNS должно быть составлено без пробелов.

Примеры:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии, используемой для подключения к Oracle.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Сноски

[1] Тайм-аут задается параметром `Timeout` в файле конфигурации (`zabbix_agent2.conf`) Zabbix агента 2 (UNIX) или Zabbix агента 2 (Windows).

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин PostgreSQL

Обзор

Параметры конфигурации позволяют настраивать поведение плагина PostgreSQL.

Рекомендуется указывать их в отдельном файле конфигурации (например, `postgresql.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы подключить этот файл в конфигурацию Zabbix agent 2.

Плагин PostgreSQL является загружаемым плагином и доступен, а также полностью описан в репозитории плагина PostgreSQL.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.PostgreSQL.CallTimeout	Установите максимальное время ожидания в секундах для завершения запроса.
Plugins.PostgreSQL.CustomQueriesPath	Укажите полный путь к каталогу, содержащему файлы `.sql` с пользовательскими запросами.
Plugins.PostgreSQL.CustomQueriesEnabled	Включение/выключение ключа элемента данных `postgresql.custom.query`.
Plugins.PostgreSQL.Default.CacheMode	Укажите режим кэширования по умолчанию для подключения PostgreSQL.
Plugins.PostgreSQL.Default.Database	Укажите имя базы данных по умолчанию.
Plugins.PostgreSQL.Default.Password	Укажите пароль по умолчанию.
Plugins.PostgreSQL.Default.TlsCaFile	Укажите полный путь к файлу, содержащему сертификат(ы) CA верхнего уровня для проверки сертификата узла.
Plugins.PostgreSQL.Default.TlsCertFile	Укажите полный путь к файлу, содержащему сертификат PostgreSQL или цепочку сертификатов.
Plugins.PostgreSQL.Default.TlsConnect	Укажите уровень шифрования по умолчанию для обмена данными между Zabbix agent 2 и контролируемыми базами данных.
Plugins.PostgreSQL.Default.TlsKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ PostgreSQL.
Plugins.PostgreSQL.Default.Uri	Укажите URI по умолчанию.

Parameter	Description
Plugins.PostgreSQL.Default.User	Укажите имя пользователя по умолчанию.
Plugins.PostgreSQL.KeepAlive	Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.
Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode	Укажите режим кэширования для именованной сессии подключения PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Укажите имя базы данных для именованной сессии.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Укажите пароль для именованной сессии.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Укажите полный путь к файлу, содержащему сертификат(ы) CA верхнего уровня для проверки сертификата узла.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат PostgreSQL или цепочку сертификатов.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Укажите уровень шифрования для именованной сессии при обмене данными между Zabbix agent 2 и контролируемыми базами данных.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.URI	Укажите URI для именованной сессии.
Plugins.PostgreSQL.Sessions.<SessionName>.User	Укажите имя пользователя для именованной сессии.

|Plugins.PostgreSQL.System.Path|Укажите путь к исполняемому файлу плагина PostgreSQL.| |Plugins.PostgreSQL.Timeout|Установите тайм-аут выполнения запроса.|

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Plugins.PostgreSQL.CallTimeout

Установите максимальное время ожидания в секундах для завершения запроса.

По умолчанию: `global timeout`
 Диапазон: `1-30`

Plugins.PostgreSQL.CustomQueriesPath

Укажите полный путь к каталогу, содержащему файлы `.sql` с пользовательскими запросами.

По умолчанию:
`/usr/local/share/zabbix/custom-queries/postgreSQL` - для систем Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` - для систем Windows (где `*` — это имя диска из переменной среды `ProgramFiles`)

Пример:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Включить выполнение ключа элемента данных `postgresql.custom.query`.
 Если отключено, запросы не будут загружаться из пути к каталогу пользовательских запросов.

По умолчанию: `false`
 Диапазон: `true/false`

Plugins.PostgreSQL.Default.CacheMode

Укажите режим кэширования по умолчанию для подключения PostgreSQL.

По умолчанию: `prepare`
 Диапазон:
`prepare` - создавать подготовленные выражения на сервере PostgreSQL;
`describe` - использовать анонимное подготовленное выражение, чтобы описать выражение без создания выражения на сервере.

Обратите внимание, что `describe` в первую очередь полезен, когда среда не позволяет использовать подготовленные выражения, например при работе через пул соединений, такой как PgBouncer.

Plugins.PostgreSQL.Default.Database

Укажите базу данных по умолчанию для подключения к PostgreSQL.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.PostgreSQL.Default.Password

Укажите пароль по умолчанию для подключения к PostgreSQL.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.PostgreSQL.Default.TLSCAFile

Укажите полный путь к файлу, содержащему сертификат(ы) корневого CA для проверки сертификата узла при зашифрованных соединениях между агентом 2 Zabbix и контролируемыми базами данных.
 Используется, если в именованной сессии не указано значение.

Обязательно: `yes`, если `Plugins.PostgreSQL.Default.TLSConnect` установлено в `verify_ca` или `verify_full`

Plugins.PostgreSQL.Default.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат PostgreSQL или цепочку сертификатов для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 Используется, если в именованной сессии не указано значение.

Обязательно: `yes`, если указан `Plugins.PostgreSQL.Default.TLSKeyFile`

Plugins.PostgreSQL.Default.TLSConnect

Укажите уровень шифрования по умолчанию для обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 Используется, если для именованной сессии не указано значение.

Диапазон:
`required` - подключение с использованием TLS в качестве режима передачи без проверки идентичности;
`verify` - подключение с использованием TLS и проверка сертификата;
`verify_full` - подключение с использованием TLS, проверка сертификата и проверка того, что идентификатор базы данных (CN), указанный в DBHost, соответствует ее сертификату.

Неуказанный тип шифрования означает незашифрованное соединение.

Plugins.PostgreSQL.Default.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ PostgreSQL для зашифрованной связи между агентом Zabbix 2 и контролируемыми базами данных.
 Используется, если в именованной сессии не указано значение.

Обязательно: `yes`, если указан `Plugins.PostgreSQL.Default.TLSCertFile`

Plugins.PostgreSQL.Default.Uri

Укажите URI по умолчанию для подключения к PostgreSQL.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: `tcp`, `unix`

Примеры:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Default.User

Укажите имя пользователя по умолчанию для подключения к PostgreSQL.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.PostgreSQL.KeepAlive

Задайте максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Укажите именованный режим кэша сеанса для подключения PostgreSQL.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

По умолчанию: `prepare`
 Диапазон:
`prepare` - создавать подготовленные операторы на сервере PostgreSQL;
`describe` - использовать анонимный подготовленный оператор для описания оператора без создания оператора на сервере.

Обратите внимание, что `describe` в первую очередь полезен, когда среда не позволяет использовать подготовленные операторы, например при работе через пул соединений, такой как PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Укажите базу данных для именованной сессии, используемой для подключения к PostgreSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии, используемой для подключения к PostgreSQL.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.
 Должен соответствовать формату пароля.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Укажите полный путь к файлу, содержащему сертификат(ы) корневого CA для проверки сертификата удаленной стороны при зашифрованных соединениях между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Обязательно: `yes`, если `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` установлено в `verify_ca` или `verify_full`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат PostgreSQL или цепочку сертификатов для зашифрованного обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Обязательно: `yes`, если указан `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Укажите именованный уровень шифрования для обмена данными между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задает имя сеанса для использования в ключах элементов данных.

Диапазон:
`required` - подключение с использованием TLS в качестве режима транспортировки без проверки идентичности;
`verify_ca` - подключение с использованием TLS и проверкой сертификата;
`verify_full` - подключение с использованием TLS, проверкой сертификата и проверкой того, что идентификатор базы данных (CN), указанный в DBHost, соответствует ее сертификату.

Неуказанный тип шифрования означает незашифрованное соединение.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Укажите полное имя пути к файлу, содержащему закрытый ключ PostgreSQL для зашифрованной связи между агентом Zabbix 2 и контролируемыми базами данных.
 <SessionName> - задайте имя сеанса для использования в ключах элементов данных.

Обязательно: `yes`, если указан `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile`.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к PostgreSQL.
 <SessionName> - задайте имя сессии для использования в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы.
 Должен соответствовать формату URI.
 Поддерживаемые схемы: `tcp`, `unix`

Примеры:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии, используемой для подключения к PostgreSQL.
 <SessionName> - задает имя сессии для использования в ключах элементов данных.

Plugins.PostgreSQL.System.Path

Укажите путь к исполняемому файлу плагина PostgreSQL.

Обязательно: `yes`

Пример:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Установите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах, после которой он будет принудительно остановлен).

По умолчанию: `global timeout`
 Диапазон: 1-30

Параметры

Parameter	Description
-V --version	Вывести версию плагина и информацию о лицензии.
-h --help	Вывести справочную информацию (сокращенный вариант).
-t, --test <item key>	Запустить плагин для тестирования (конфигурация плагина игнорируется).

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин Redis

Обзор

Параметры конфигурации позволяют настраивать поведение плагина Redis.

Рекомендуется указывать их в отдельном файле конфигурации (например, `redis.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, нажмите на его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Redis.Default.Password	Укажите пароль по умолчанию для подключения к Redis.
Plugins.Redis.Default.TLSConnect	Укажите уровень шифрования, требуемый для соединений Zabbix agent 2.
Plugins.Redis.Default.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA.
Plugins.Redis.Default.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента.
Plugins.Redis.Default.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ агента.
Plugins.Redis.Default.Uri	Укажите URI по умолчанию для подключения к Redis.
Plugins.Redis.Default.User	Укажите имя пользователя по умолчанию для подключения к Redis.
Plugins.Redis.KeepAlive	Задайте максимальное время ожидания (в секундах) перед закрытием неиспользуемых соединений плагина.
Plugins.Redis.Sessions.<SessionName>.Password	Укажите пароль для именованной сессии подключения к Redis.
Plugins.Redis.Sessions.<SessionName>.TLSConnect	Укажите уровень шифрования, требуемый для соединений Zabbix agent 2 для именованной сессии.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	Укажите полный путь к файлу, содержащему сертификаты корневого CA для именованной сессии.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	Укажите полный путь к файлу, содержащему сертификат агента для именованной сессии.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	Укажите полный путь к файлу, содержащему закрытый ключ агента для именованной сессии.
Plugins.Redis.Sessions.<SessionName>.Uri	Укажите URI именованной сессии для подключения к Redis.
Plugins.Redis.Sessions.<SessionName>.User	Укажите имя пользователя именованной сессии для подключения к Redis.
Plugins.Redis.Timeout	Задайте тайм-аут выполнения запроса.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные среды.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Сведения о параметре

Plugins.Redis.Default.Password

Укажите пароль по умолчанию для подключения к Redis.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

Plugins.Redis.Default.TLSConnect

Укажите уровень шифрования, требуемый для обмена данными агента 2 Zabbix.
 Пустое или не заданное значение считается отсутствием TLS.

Диапазон: `required` - зашифровано, без проверки (только для тестирования), `verify_ca` - зашифровано, сертификат сервера, `verify_full` - зашифровано, сертификат сервера и имя сервера проверяются по SAN

Plugins.Redis.Default.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты корневого CA.

Plugins.Redis.Default.TLSCertFile

Укажите полный путь к файлу, содержащему сертификат агента.

Plugins.Redis.Default.TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента.

Plugins.Redis.Default.Uri

Укажите URI по умолчанию для подключения к Redis.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.
 Встроенные учетные данные будут проигнорированы. Должен соответствовать формату URI.
 Поддерживаемые схемы: `tcp`, `unix`
 Схему можно опустить.
 Порт можно опустить (по умолчанию=6379).

По умолчанию: `tcp://localhost:6379`

Примеры:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Укажите имя пользователя по умолчанию для подключения к Redis.
 Используется, если значение не указано в ключе элемента данных или именованной сессии.

По умолчанию: `default`

Plugins.Redis.KeepAlive

Установите максимальное время ожидания (в секундах) перед закрытием неиспользуемых подключений плагина.

По умолчанию: 300
 Диапазон: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Укажите пароль для именованной сессии, используемый для подключения к Redis.
 <SessionName> - задает имя сессии, которое будет использоваться в ключах элементов данных.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Укажите уровень шифрования, требуемый для коммуникаций агента Zabbix 2 для именованной сессии.
 <SessionName> - задает имя сессии, которое будет использоваться в ключах элементов данных.
 Пустое или не заданное значение считается отсутствием TLS.

Диапазон: `required` - зашифровано, без проверки (только для тестирования), `verify_ca` - зашифровано, сертификат сервера, `verify_full` - зашифровано, сертификат сервера и имя сервера проверяются с помощью SAN

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Укажите полный путь к файлу, содержащему сертификаты CA верхнего уровня для именованной сессии.
 <SessionName> - задает имя сессии, которое будет использоваться в ключах элементов данных.

Обязательно: `yes`, если `Plugins.Redis.Sessions.<SessionName>.TLSConnect` имеет значение `verify_ca` или `verify_full`

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Укажите полное имя пути к файлу, содержащему сертификат агента для именованной сессии.
 <SessionName> - задайте имя сессии, которое будет использоваться в ключах элементов данных.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Укажите полный путь к файлу, содержащему закрытый ключ агента для именованной сессии.
 <SessionName> - задайте имя сессии, которое будет использоваться в ключах элементов данных.

Plugins.Redis.Sessions.<SessionName>.Uri

Укажите URI именованной сессии для подключения к Redis.
 <SessionName> - задает имя сессии, которое будет использоваться в ключах элементов данных.
 Встроенные учетные данные будут проигнорированы. Должен соответствовать формату URI.
 Поддерживаемые схемы: `tcp`, `unix`
 Схему можно опустить.
 Порт можно опустить (по умолчанию=6379).

По умолчанию: `tcp://localhost:6379`

Примеры:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Укажите имя пользователя для именованной сессии, используемое для подключения к Redis.
 <SessionName> - задает имя сессии, которое будет использоваться в ключах элементов данных.

Plugins.Redis.Timeout

Установите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах перед его принудительным завершением).

По умолчанию: `global timeout`
 Диапазон: 1-30

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Плагин SMART

Обзор

Параметры конфигурации позволяют настраивать поведение плагина SMART.

Рекомендуется указывать их в отдельном файле конфигурации (например, `smart.conf`), а не в файле конфигурации агента 2. Затем используйте директиву Include, чтобы включить этот файл в конфигурацию Zabbix agent 2.

Все параметры являются **необязательными**, если явно не указано, что параметр обязателен.

Эта страница имеет следующую структуру:

- Обзор параметров (чтобы увидеть полные сведения о параметре, щелкните его имя)
- Сведения о параметрах

Parameter	Description
Plugins.Smart.Path	Укажите путь к исполняемому файлу smartctl.
Plugins.Smart.Timeout	Задайте тайм-аут выполнения запроса (время ожидания завершения запроса в секундах перед его остановкой).

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Путь к исполняемому файлу `smartctl` должен быть задан либо путем добавления его каталога в системную переменную среды `PATH`, либо путем настройки `Plugins.Smart.Path`; это относится как к Linux, так и к Windows;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Сведения о параметре

Plugins.Smart.Path

Укажите путь к исполняемому файлу smartctl.

По умолчанию: `smartctl`

Plugins.Smart.Timeout

Установите тайм-аут выполнения запроса (продолжительность ожидания завершения запроса в секундах, по истечении которой он будет принудительно остановлен).

По умолчанию: `global timeout<br>` Диапазон: 1–30

См. также

- Описание общих параметров конфигурации Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Инструкции по настройке плагинов

Протокол плагина Zabbix agent 2

Протокол Zabbix agent 2 основан на коде, размере и модели данных.

Код

Type	Size	Comments
Byte	4	Тип полезной нагрузки, в настоящее время поддерживается только JSON.

Размер

Type	Size	Comments
Byte	4	Размер текущей полезной нагрузки в байтах.

Данные полезной нагрузки

Type	Size	Comments
Byte	Определяется полем *Size*	Данные в формате JSON.

Определение данных полезной нагрузки

Общие данные

Эти параметры присутствуют во всех запросах/ответах:

Name	Type	Comments
id	uint32	Для запросов - возрастающий идентификатор, используемый для связывания запросов с ответами. Уникален в пределах одного направления запроса (то есть от агента к плагину или от плагина к агенту). Для ответов - ID соответствующего запроса.
type	uint32	Тип запроса.

Запрос журнала

Запрос, отправляемый плагином для записи сообщения журнала в файл журнала агента.

direction	plugin → агент
response	no

Параметры, специфичные для запросов журнала:

Name	Type	Comments
severity	uint32	Уровень серьезности сообщения (уровень журнала).
message	string	Сообщение, которое нужно записать в журнал.

Пример:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Запрос регистрации

Запрос, отправляемый агентом на этапе запуска агента для получения предоставляемых метрик и регистрации плагина.

direction	агент → plugin
response	yes

Параметры, специфичные для запросов регистрации:

Name	Type	Comments
version	string	Версия протокола <major>.<minor>

Пример:

```
{"id":1,"type":2,"version":"1.0"}
```

Ответ на register

Ответ плагина на запрос register.

direction	plugin → агент
response	n/a

Параметры, специфичные для ответов register:

Name	Type	Comments
name	string	Имя плагина.
metrics	array of strings (optional)	Метрики с описаниями, как они используются в плагине. Возвращает RegisterMetrics(). Отсутствует, если возвращается ошибка.
interfaces	uint32 (optional)	Битовая маска поддерживаемых плагином интерфейсов. Отсутствует, если возвращается ошибка.
error	string (optional)	Сообщение об ошибке, возвращаемое, если плагин не может быть запущен. Отсутствует, если возвращаются метрики.

Примеры:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

или

```
{"id":2,"type":3,"error":"error message"}
```

Запрос Start

Запрос на выполнение функции Start интерфейса Runner.

direction	агент → plugin
response	no

Запрос не содержит специальных параметров, он включает только параметры общих данных.

Пример:

```
{"id":3,"type":4}
```

Завершить запрос

Запрос, отправляемый агентом для завершения работы плагина.

direction	агент → plugin
response	нет

Запрос не имеет специальных параметров, он содержит только параметры общих данных.

Пример:

```
{"id":3,"type":5}
```

Запрос на экспорт

Запрос на выполнение функции Export интерфейса Exporter.

direction	агент → plugin
response	no

Параметры, специфичные для запросов на экспорт:

Name	Type	Comments
key	string	Ключ плагина.
parameters	array of strings (optional)	Параметры для функции Export.

Пример:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Ответ экспорта

Ответ от функции Export интерфейса Exporter.

direction	plugin → agent
response	n/a

Параметры, специфичные для ответов экспорта:

Name	Type	Comments
value	string (optional)	Значение ответа от функции Export. Отсутствует, если возвращена ошибка.
error	string (optional)	Сообщение об ошибке, если функция Export не была выполнена успешно. Отсутствует, если возвращено значение.

Примеры:

```
{"id":5,"type":7,"value":"response"}
```

или

```
{"id":5,"type":7,"error":"error message"}
```

Настройка запроса

Запрос на выполнение функции *Configure* интерфейса *Configurator*.

direction	агент → plugin
response	n/a

Параметры, специфичные для запросов *Configure*:

Name	Type	Comments
global_options	JSON object	JSON object, содержащий глобальные параметры конфигурации агента.
private_options	JSON object (optional)	JSON object, содержащий частные параметры конфигурации plugin, если они указаны.

Пример:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Проверка запроса

Запрос на выполнение функции *Validate* интерфейса *Configurator*.

direction	агент → plugin
response	yes

Параметры, специфичные для запросов *Validate*:

Name	Type	Comments
private_options	JSON object (optional)	JSON object, содержащий приватные параметры конфигурации плагина, если они предоставлены.

Пример:

```
{"id":7,"type":9,"private_options":{...}}
```

Проверка ответа

Ответ функции *Validate* интерфейса *Configurator*.

direction	plugin → агент
response	n/a

Параметры, специфичные для ответов *Validate*:

Name	Type	Comments
error	string (optional)	Сообщение об ошибке, возвращаемое, если функция Validate не выполнена успешно. Отсутствует при успешном выполнении.

Пример:

```
{"id":8,"type":10}
```

или

```
{"id":8,"type":10,"error":"error message"}
```

Java gateway

Обзор Zabbix Java gateway можно установить из исходного кода или пакетов.

Встроенная поддержка мониторинга приложений JMX реализована в виде демона Zabbix под названием «Zabbix Java gateway». Zabbix Java gateway — это демон, написанный на Java. Чтобы получить значение определенного счетчика JMX на узле сети, сервер Zabbix запрашивает Zabbix Java gateway, который использует API управления JMX для удаленного запроса нужного приложения. В приложении не требуется устанавливать дополнительное программное обеспечение — его необходимо только запустить с параметром `-Dcom.sun.management.jmxremote` в командной строке.

Java gateway принимает входящие подключения от сервера Zabbix или прокси и может использоваться только как «пассивный прокси». В отличие от прокси Zabbix, он также может использоваться прокси Zabbix (прокси Zabbix нельзя объединять в цепочку). Доступ к каждому Java gateway настраивается непосредственно в конфигурационном файле Zabbix сервера или Zabbix прокси, поэтому для каждого сервера Zabbix или прокси Zabbix можно настроить только один Java gateway. Если на узле сети есть элементы данных типа *JMX agent* и элементы данных других типов, для получения данных в Java gateway будут передаваться только элементы данных типа *JMX agent*.

Если элемент данных необходимо обновить через Java gateway, сервер или прокси Zabbix подключается к Java gateway и запрашивает значение, которое Java gateway, в свою очередь, получает и передаёт обратно серверу или прокси. Таким образом, Java gateway не кэширует значения.

Сервер или прокси Zabbix имеет специальный тип процессов, подключающихся к Java gateway; их количество регулируется параметром `StartJavaPollers`. Внутри Java gateway запускается несколько потоков, количество которых регулируется параметром `START_POLLERS`. На стороне сервера, если подключение занимает более "Timeout" секунд, оно будет прервано, однако Java gateway может по-прежнему получать значение из счётчика JMX. Для решения этой проблемы в Java gateway предусмотрен параметр `TIMEOUT`, позволяющий задать время ожидания для сетевых операций JMX.

См. также:

- Параметры конфигурации: Zabbix Java gateway
- Установка из пакетов Debian/Ubuntu
- Установка из пакетов RHEL
- Установка из исходного кода

Для достижения лучшей производительности Zabbix сервер и прокси будут пытаться максимально объединить запросы к одной цели JMX (зависит от интервалов обновления элементов данных) и отправлять их в Java gateway за одно подключение.

Рекомендуется выставить значение `StartJavaPollers` меньшим или равным `START\_POLLERS`, в противном случае могут возникнуть ситуации, когда потоков Java gateway может не хватить для обслуживания входящих запросов. В таких случаях Java gateway использует `ThreadPoolExecutor.CallerRunsPolicy`, что означает, что основной поток будет обрабатывать входящий запрос и временно не будет принимать никакие новые запросы.

Если Вы пытаетесь посредством Zabbix Java gateway контролировать Java приложения на базе Wildfly, то установите, пожалуйста, последний `jboss-client.jar`, который доступен на странице загрузки Wildfly [en].

Параметры конфигурации: Zabbix Java gateway

Если вы используете скрипты `startup.sh` и `shutdown.sh` для запуска Zabbix Java gateway, то необходимые параметры конфигурации можно указать в файле `settings.sh`. Скрипты запуска и завершения работы подключают файл настроек и выполняют преобразование переменных оболочки (перечисленных в первом столбце) в свойства Java (перечисленные во втором столбце).

Если вы запускаете Zabbix Java gateway вручную, напрямую выполняя `java`, то соответствующие свойства Java указываются в командной строке.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	IP-адрес для прослушивания.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Порт для прослушивания.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Имя файла PID. Если не указано, Zabbix Java Gateway запускается как консольное приложение.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	no			Имя файла свойств. Может использоваться для задания дополнительных свойств в формате ключ-значение так, чтобы они не были видны в командной строке, или для переопределения существующих. Например: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Количество рабочих потоков для запуска.
TIMEOUT	zabbix.timeout	no	1-30	3	Время ожидания сетевых операций (в секундах).

Warning:
Порт 10052 не зарегистрирован IANA.

Установка из исходных кодов

Обзор

Если установлен из исходных кодов, следующая информация поможет вам в настройке Zabbix Java gateway.

Обзор файлов

Если вы получили Java gateway из исходного кода, в `$PREFIX/sbin/zabbix_java` у вас должен был появиться набор скриптов оболочки, JAR-файлов и файлов конфигурации. Назначение этих файлов описано ниже.

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

Сам JAR-файл Java gateway.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Зависимости Java gateway: библиотеки Logback, SLF4J и Android JSON.

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

Файлы конфигурации Logback.

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

Вспомогательные скрипты для запуска и остановки Java gateway.

`settings.sh`

Файл конфигурации, который подключается приведенными выше скриптами запуска и остановки.

Настройка и запуск Java gateway

По умолчанию Java gateway прослушивает порт 10052. Если вы планируете запускать Java gateway на другом порту, вы можете указать это в скрипте `settings.sh`. См. описание файла конфигурации Java gateway, чтобы узнать, как задать этот и другие параметры.

Warning:
Порт 10052 не зарегистрирован в IANA.

После того как вы настроите параметры, вы можете запустить Java gateway, выполнив скрипт `startup.sh`:

```
./startup.sh
```

Аналогично, когда Java gateway больше не нужен, выполните скрипт `shutdown.sh`, чтобы остановить его:

```
./shutdown.sh
```

Обратите внимание, что в отличие от сервера или прокси, Java gateway является легковесным и не требует базы данных.

Настройка сервера для использования с Java gateway

После запуска Java gateway необходимо указать серверу Zabbix, где найти Zabbix Java gateway. Для этого в файле конфигурации сервера задаются параметры `JavaGateway` и `JavaGatewayPort`. Если узел сети, на котором работает приложение JMX, контролируется через прокси Zabbix, то параметры подключения следует указывать в файле конфигурации прокси.

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

По умолчанию сервер не запускает никаких процессов, связанных с мониторингом JMX. Однако если вы хотите использовать его, необходимо указать количество предварительно порожденных экземпляров Java pollers. Это делается так же, как и при указании обычных pollers и trappers.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после завершения настройки.

Отладка Java gateway

В случае возникновения каких-либо проблем с Java gateway или в случае, если сообщение об ошибке элемента данных в веб-интерфейсе недостаточно информативно, вы можете обратиться к файлу журнала Java gateway.

По умолчанию Java gateway записывает журнал в файл `/tmp/zabbix_java.log` с уровнем журналирования «info». Бывает, что этой информации недостаточно и требуется информация уровня журналирования «debug». Чтобы увеличить уровень журналирования, отредактируйте файл `lib/logback.xml` и измените атрибут level тега `<root>` на значение `debug`:

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

Обратите внимание, что в отличие от Zabbix сервера или Zabbix прокси, вам не нужно перезапускать Zabbix Java gateway после изменения файла `logback.xml` — изменения в файле `logback.xml` подхватятся автоматически. Когда вы завершите отладку, вы можете вернуть уровень журналирования обратно в «info».

Если вы хотите записывать журнал в другой файл или на совершенно другой носитель, такой как база данных, настройте файл `logback.xml` в соответствии с вашими потребностями. Обратитесь к Руководству по Logback [en] для получения более подробных сведений.

Иногда в целях отладки полезно запустить Java gateway как консольное приложение, а не как демон. Чтобы это сделать, закомментируйте переменную `PID_FILE` в `settings.sh`. Если `PID_FILE` не указан, скрипт `startup.sh` запускает Java gateway как консольное приложение, при этом Logback использует файл `lib/logback-console.xml`, который не только выводит журнал в консоль, но и имеет уровень журналирования «debug».

В заключение, отметим, поскольку Java gateway использует для журналирования SLF4J, вы можете заменить Logback выбранным вами фреймворком, поместив соответствующий JAR файл в каталог lib. Обратитесь к Руководству по SLF4 [en] для получения более подробных сведений.

Мониторинг JMX

Смотрите страницу JMX мониторинга для получения более подробных сведений.

Установка из пакетов Debian/Ubuntu

Обзор

Если установлено из пакетов, следующая информация поможет вам в настройке Zabbix Java gateway.

Настройка и запуск Java gateway

Конфигурацию Java gateway можно изменить в файле:

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

Подробнее см. параметры конфигурации Zabbix Java gateway.

Чтобы запустить Zabbix Java gateway:

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

Чтобы автоматически запускать Zabbix Java gateway при загрузке системы:

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

Настройка сервера для использования Java gateway

После запуска Java gateway необходимо указать серверу Zabbix, где найти Zabbix Java gateway. Для этого в файле конфигурации сервера задаются параметры `JavaGateway` и `JavaGatewayPort`. Если узел сети, на котором работает JMX-приложение, контролируется прокси Zabbix, то вместо этого параметры подключения указываются в файле конфигурации прокси.

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

По умолчанию сервер не запускает никаких процессов, связанных с мониторингом JMX. Однако если вы хотите использовать его, необходимо указать количество предварительно порожденных экземпляров Java pollers. Это делается так же, как и при указании обычных pollers и trappers.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после завершения настройки.

Отладка Java gateway

Файл журнала Zabbix Java gateway:

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

Если вы желаете увеличить уровень журналирования, отредактируйте следующий файл:

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

и измените `info` в атрибуте `level="info"` на `debug` или даже `trace` (для более глубокой отладки):

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

</configuration>
```

Мониторинг JMX

Смотрите страницу мониторинга JMX для получения более подробных сведений.

Установка из пакетов RHEL

Обзор

Если установлено из пакетов, следующая информация поможет вам в настройке Zabbix Java gateway.

Настройка и запуск Java gateway

Параметры конфигурации Zabbix Java gateway можно изменить в файле:

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

Подробнее см. параметры конфигурации Zabbix Java gateway.

Чтобы запустить Zabbix Java gateway:

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

Чтобы автоматически запускать Zabbix Java gateway при загрузке системы:

RHEL 7 и новее:

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

RHEL до 7:

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

Настройка сервера для использования с Java gateway

После запуска Java gateway необходимо указать Zabbix серверу, где найти Zabbix Java gateway. Для этого в файле конфигурации сервера задаются параметры `JavaGateway` и `JavaGatewayPort`. Если узел сети, на котором запущено JMX-приложение, контролируется Zabbix прокси, то параметры подключения следует указать в файле конфигурации прокси.

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

По умолчанию сервер не запускает никаких процессов, связанных с мониторингом JMX. Однако если вы хотите использовать его, необходимо указать количество предварительно порожденных экземпляров Java pollers. Это делается так же, как и при указании обычных pollers и trappers.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после завершения настройки.

Отладка Java gateway

Файл журнала Zabbix Java gateway:

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

Если вы желаете увеличить уровень журналирования, отредактируйте следующий файл:

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

и измените `info` в атрибуте `level="info"` на `debug` или даже `trace` (для более глубокой отладки):

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

</configuration>
```

Мониторинг JMX

Смотрите страницу JMX мониторинга для получения более подробных сведений.

Отправитель

Обзор Zabbix sender — это утилита командной строки, которую можно использовать для отправки данных о производительности на сервер Zabbix или прокси для обработки.

Обычно утилита используется в длительно выполняющихся пользовательских скриптах для периодической отправки данных о доступности и производительности.

Для непосредственной отправки результатов на сервер Zabbix или прокси необходимо настроить элемент данных типа *Zabbix trapper*.

См. также:

- Протокол Zabbix sender
- Динамическая библиотека Zabbix sender для Windows
- Библиотека Python для Zabbix, содержащая встроенные функции для работы в качестве Zabbix sender

Запуск Zabbix sender Пример запуска UNIX-версии Zabbix sender:

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

где:

- `z` - узел сети сервера Zabbix (также можно использовать IP-адрес);
- `s` - техническое имя отслеживаемого узла сети (зарегистрированное в веб-интерфейсе Zabbix);
- `k` - ключ элемента данных;
- `o` - отправляемое значение.

Attention:
Параметры, содержащие пробелы, необходимо заключать в двойные кавычки.

Zabbix sender можно использовать для отправки нескольких значений из входного файла. Дополнительную информацию см. на странице руководства Zabbix sender.

Если указан файл конфигурации, Zabbix sender использует все адреса, определенные в параметре конфигурации агента `ServerActive`, для отправки данных. Если отправка на один адрес завершается ошибкой, sender пытается отправить данные на другие адреса. Если отправка пакета данных на один адрес завершается ошибкой, следующие пакеты на этот адрес не отправляются.

Zabbix sender принимает строки в кодировке UTF-8 (как в Unix-подобных системах, так и в Windows), если в начале файла отсутствует метка порядка байтов (BOM).

Zabbix sender в Windows можно запускать аналогичным образом:

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

В сценариях отправки в реальном времени `zabbix_sender` собирает несколько значений, переданных ему через короткие промежутки времени, и отправляет их серверу через одно соединение. Значение, полученное не более чем через 0,2 секунды после предыдущего, может быть помещено в тот же стек, однако максимальное время опроса по-прежнему составляет 1 секунду.

Note:
Zabbix sender завершит работу, если в указанном файле конфигурации присутствует недопустимая запись параметра (не соответствующая формату `parameter=value`).

Запуск Zabbix sender с низкоуровневым обнаружением Пример запуска Zabbix sender для отправки значения в формате JSON для низкоуровневого обнаружения:

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

Чтобы это работало, правило низкоуровневого обнаружения должно иметь тип элемента данных *Zabbix trapper* (в этом примере — с ключом `trapper.discovery.item`).

Протокол Zabbix sender

Сервер Zabbix и прокси Zabbix используют основанный на JSON протокол обмена для получения данных от Zabbix sender. Данные могут быть получены с помощью элемента данных trapper или элемента данных HTTP agent с включенным trapping.

Сообщения запроса и ответа должны начинаться с заголовка и длины данных.

Запрос Zabbix sender

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Ответ сервера Zabbix

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Запрос Zabbix sender с временной меткой

Кроме того, Zabbix sender может отправлять запрос с временной меткой и наносекундами.

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Ответ сервера Zabbix

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Динамическая библиотека Zabbix sender для Windows

В среде Windows приложения могут отправлять данные на сервер/прокси Zabbix с помощью динамической библиотеки Zabbix sender (`zabbix_sender.dll`) вместо запуска внешнего процесса (`zabbix_sender.exe`).

Для компиляции пользовательских приложений с `zabbix_sender.dll` требуются `zabbix_sender.h` и `zabbix_sender.lib`.

Получение

Есть два способа получить `zabbix_sender.dll`.

1. Скачайте файлы `zabbix_sender.h`, `zabbix_sender.lib` и `zabbix_sender.dll` в виде ZIP-архива.

При выборе параметров загрузки обязательно укажите "No encryption" в разделе *Encryption* и "Archive" в разделе *Packaging*. Затем скачайте Zabbix agent (не Zabbix agent 2).

Файлы `zabbix_sender.h`, `zabbix_sender.lib` и `zabbix_sender.dll` будут находиться в загруженном ZIP-архиве в каталоге `bin\dev`. Распакуйте файлы туда, где они вам нужны.

2. Соберите `zabbix_sender.dll` из исходного кода (см. инструкции).

Динамическая библиотека с файлами для разработки будет находиться в каталоге `bin\winXX\dev`. Чтобы использовать ее, подключите заголовочный файл `zabbix_sender.h` и выполните линковку с библиотекой `zabbix_sender.lib`.

См. также

- пример простой утилиты Zabbix sender, реализованной с использованием динамической библиотеки Zabbix sender, чтобы показать, как использовать библиотеку;
- файл zabbix_sender.h с интерфейсными функциями динамической библиотеки Zabbix sender. Этот файл содержит документацию, объясняющую назначение каждой интерфейсной функции, ее аргументы и возвращаемое значение.

Получить

Обзор Zabbix get — это утилита командной строки, которая может использоваться для взаимодействия с агентом Zabbix и получения необходимой информации от агента.

Обычно эта утилита используется для устранения неполадок в агентах Zabbix.

См. также библиотеку Python для Zabbix, в которой встроена функциональность, позволяющая работать как Zabbix get.

Запуск Zabbix get Пример запуска Zabbix get в UNIX для получения значения загрузки процессора от агента:

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

Еще один пример запуска Zabbix get для извлечения строки с веб-сайта:

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

Обратите внимание, что ключ элемента данных здесь содержит пробел, поэтому для оболочки ключ элемента данных заключен в кавычки. Кавычки не являются частью ключа элемента данных; оболочка удалит их, и они не будут переданы агенту Zabbix.

Если ключ элемента данных не поддерживается, Zabbix get вернет код выхода 1.

Zabbix get принимает следующие параметры командной строки:

```
-s --host <host name or IP>                              IP-
-p --port <port number>                                 ,              (       : 10050)
-I --source-address <IP address>                  IP-
-t --timeout <seconds>                              -  .           : 1-600     (       : 30    )
-k --key <item key>                                     ,
-P --protocol <value>                             ,                    .     :
                                                  auto -                JSON-     ,
                                                  json -                JSON-
                                                  plaintext -                          ,
-h --help
-V --version

--tls-connect <value>                                          .     :
                                                  unencrypted -               (       )
                                                  psk -                 TLS
                                                  cert -                 TLS
--tls-ca-file <CA file>                                 ,                   CA( )
--tls-crl-file <CRL file>                               ,
--tls-agent-cert-issuer <cert issuer>
--tls-agent-cert-subject <cert subject>
--tls-cert-file <cert file>                             ,
--tls-key-file <key file>                               ,
--tls-psk-identity <PSK-identity>                              ,
--tls-psk-file <PSK-file>                               ,
--tls-cipher13 <cipher-string>                    OpenSSL 1.1.1          TLS 1.3.
--tls-cipher <cipher-string>                      GnuTLS (   TLS 1.2     )           OpenSSL (      TLS
```

Дополнительные сведения см. в man-странице Zabbix get.

Zabbix get в Windows можно запускать аналогичным образом:

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

JS

Обзор zabbix_js — это утилита командной строки, которую можно использовать для тестирования встроенных скриптов.

Эта утилита выполнит пользовательский скрипт со строковым параметром и выведет результат. Скрипты выполняются с использованием встроенного механизма скриптов Zabbix.

В случае ошибок компиляции или выполнения zabbix_js выведет сообщение об ошибке в stderr и завершит работу с кодом 1.

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

Использование zabbix_js принимает следующие параметры командной строки:

```
-s, --script script-file
                                                    "-",                stdin
-i, --input input-file

                                              -p
                                                    "-",                stdin
-p, --param input-param                         :       ,
                                              -i
-l, --loglevel log-level
                                        : 0-5
-t, --timeout timeout                     -
                                            : 1-600     (       : 10    )
-h, --help
-V, --version
-w <webdriver url>
```

Примеры: Пример 1: Выполнение скрипта с переменной, содержащей значение

Скрипт (script-file.js):

```
return value;
```

Использование:

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

Вывод: example_value

Пример 2: Выполнение скрипта с файлом, содержащим входные данные

Файл со скриптом (script-file.js):

```
return value;
```

Файл с входными данными (example.txt):

```
```

Использование:

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

Вывод:

Пример 3: Выполнение скрипта с вводом, прочитанным из stdin

Файл со скриптом (script-file.js):

```
return value;
```

Использование:

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

Ввод (stdin):

```
          stdin
```

Вывод: stdin

Веб-сервис

Обзор Веб-сервис Zabbix — это процесс, который используется для взаимодействия с внешними веб-сервисами. В настоящее время веб-сервис Zabbix используется для создания и отправки запланированных отчетов, а в будущем планируется добавить дополнительную функциональность.

Сервер Zabbix подключается к веб-сервису по HTTP(S). Для веб-сервиса Zabbix требуется, чтобы Google Chrome был установлен на том же узле сети; в некоторых дистрибутивах служба также может работать с Chromium (см. известные проблемы).

Установка Официальный пакет zabbix-web-service доступен в репозитории Zabbix.

Чтобы скомпилировать веб-сервис Zabbix из исходного кода, укажите параметр конфигурации `--enable-webservice`.

Чтобы настроить веб-сервис Zabbix, обновите параметры файла конфигурации веб-сервиса.

Attention:
Настоятельно рекомендуется настроить шифрование между сервером Zabbix и веб-сервисом Zabbix с помощью сертификатов. По умолчанию данные, передаваемые между сервером Zabbix и веб-сервисом Zabbix, не шифруются, что может привести к несанкционированному доступу.

Параметры конфигурации: Zabbix web service

Обзор

Веб-служба Zabbix — это процесс, который используется для взаимодействия с внешними веб-службами.

Параметры, поддерживаемые файлом конфигурации веб-службы Zabbix (`zabbix_web_service.conf`), перечислены в этом разделе.

Параметры перечислены без дополнительной информации. Щелкните параметр, чтобы увидеть полные сведения.

Parameter	Description
AllowedIP	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix и прокси Zabbix.
DebugLevel	Уровень отладки.
IgnoreURLCertErrors	Определяет обработку ошибок проверки TLS-сертификата при обращении к URL веб-интерфейса.
Include	Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации.
ListenPort	Служба будет прослушивать этот порт для подключений от сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogType	Тип вывода журнала.
Timeout	Максимальное время (в секундах), затрачиваемое на формирование PDF отчета панели мониторинга.
TLSAccept	Какие входящие подключения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневого CA, используемые для проверки сертификата узла связи при защищенном обмене данными между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат службы или цепочку сертификатов, используемые для защищенного обмена данными между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ службы, используемый для защищенного обмена данными между компонентами Zabbix.

Все параметры являются необязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные среды;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметре

AllowedIP

Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix и прокси Zabbix. Входящие подключения будут приниматься только от узлов сети, перечисленных здесь.
Если поддержка IPv6 включена, то `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются эквивалентными, а `::/0` позволит использовать любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать, чтобы разрешить любой IPv4-адрес.

Пример:

`127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com`

Обязательно: да

DebugLevel

Укажите уровень отладки: - *0* - базовая информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - для отладки (выводит много информации); - *5* - расширенная отладка (выводит еще больше информации).

По умолчанию: 3
 Диапазон: 0-5

IgnoreURLCertErrors

Определяет обработку ошибок проверки TLS-сертификата при обращении к URL веб-интерфейса:
0 - не игнорировать ошибки сертификата;
1 - игнорировать ошибки сертификата.

По умолчанию: 0
 Диапазон: 0-1

Include

Укажите каталог или отдельные файлы, которые нужно включить в файл конфигурации.
 Во время установки Zabbix этот параметр устанавливается в `/usr/local/etc`, если вы не измените его на этапе компиляции.

Общие правила для параметра `Include`:

- Путь может быть относительным к расположению файла `zabbix_web_service.conf`.
- В одном файле конфигурации можно указать несколько параметров `Include`.
- Zabbix обрабатывает параметр `Include` сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для включаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила для включения каталогов или шаблонов файлов:

- Не следует предполагать какой-либо порядок включения (например, файлы не включаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном включаемом файле (например, чтобы переопределить общее значение более конкретным), так как невозможно предсказать, какое значение будет использовано.
- Обратите внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан `include/my_specific.conf.BAK`), поскольку будут включены оба файла.

Чтобы включить только нужные файлы из каталога, используйте подстановочный знак (`*`) для сопоставления шаблонов.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Служба будет прослушивать этот порт для подключений от сервера.

По умолчанию: 10053
 Диапазон: 1024-32767

LogFile

Имя файла журнала.

Пример:

`/tmp/zabbix_web_service.log`

Обязательно: да, если LogType установлено в `file`; в противном случае — нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 - отключить автоматическую ротацию журнала.
Примечание: если достигается ограничение размера файла журнала и ротация файла по какой-либо причине не выполняется, существующий файл журнала усекается и создается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogType

Тип вывода журнала: - *file* - записывать журнал в файл, указанный параметром LogFile; - *system* - записывать журнал в syslog; - *console* - записывать журнал в стандартный вывод.

По умолчанию: `file`

Тайм-аут

Максимальное время (в секундах), затрачиваемое на форматирование PDF отчета панели мониторинга.

По умолчанию: 10
 Диапазон: 1-30

TLSAccept

Какие входящие подключения принимать: - *unencrypted* - принимать подключения без шифрования (по умолчанию); - *cert* - принимать подключения с TLS и сертификатом.

По умолчанию: `unencrypted`

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневого CA для проверки сертификата узла-пира, используемому для зашифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат службы или цепочку сертификатов, используемый для зашифрованной связи с компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ службы, используемый для зашифрованной связи между компонентами Zabbix.

MCP-сервер

Обзор MCP-сервер Zabbix — это процесс, который позволяет ИИ-агентам работать с Zabbix с использованием протокола контекста модели (MCP). MCP-сервер взаимодействует с MCP-клиентами и API Zabbix.

С помощью MCP-сервера ИИ-агент может:

- получать группы узлов сети, группы шаблонов, узлы сети, шаблоны, элементы данных, правила LLD, интерфейсы, макросы, триггеры, проблемы, периоды обслуживания и данные истории элементов данных (за исключением элементов данных двоичного типа);
- выполнять обновление проблем: добавлять комментарии, подтверждать или снимать подтверждение, изменять важность, подавлять или отменять подавление, а также закрывать проблемы;
- создавать, обновлять и удалять периоды обслуживания.

Для авторизации используются токены API Zabbix; токены отправляются с каждым запросом. Доступ ограничен разрешениями владельца токена: видимы только данные из групп узлов сети, доступных этому пользователю, а действия, которые пользователю запрещено выполнять, отклоняются.

MCP-сервер не сохраняет состояние: между запросами сеанс не сохраняется.

> **Note:**
> Хотя MCP-сервер Zabbix может работать со старыми или более новыми версиями Zabbix, официально поддерживается только совместимость с той же версией Zabbix, что и версия самого MCP-сервера.

См. также:

- Параметры конфигурации: MCP-сервер
- Инструменты и ресурсы

Установка Официальный пакет zabbix-mcp-server доступен в репозитории Zabbix.

Чтобы установить сервер Zabbix MCP из исходного кода, соберите его с помощью `make`.

Чтобы настроить сервер Zabbix MCP, измените параметры файла конфигурации сервера MCP. Необходимо задать как минимум параметры `FrontendURL` и `AllowedIP`.

Файл конфигурации по умолчанию (`zabbix_mcp_server.conf`) находится в `/etc/zabbix/`; в нем запрещены инструменты, которые чаще всего раскрывают конфиденциальные данные. Чтобы включить инструмент, удалите соответствующую строку `DenyTool`.

См. также: Рекомендации по обеспечению безопасности

Запуск Чтобы запустить MCP-сервер, выполните:

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

Чтобы проверить файл конфигурации и завершить работу без запуска сервера, используйте параметр `-T`:

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

Поддерживаются следующие параметры командной строки:

Параметр	Описание
`-c <file>`	Путь к файлу конфигурации. По умолчанию используется `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Проверить файл конфигурации и завершить работу.
`-h`	Отобразить справочную информацию.
`-V`	Отобразить номер версии.

Если MCP-сервер установлен из исходного кода, файл модуля systemd необходимо создать вручную, например:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Подключение клиента MCP К серверу MCP можно подключить любой клиент MCP, поддерживающий транспорт Streamable HTTP. Требуются следующие настройки:

Настройка	Значение
Транспорт	Streamable HTTP
URL	`https://<host>:<ListenPort>/mcp`
Заголовок	`Authorization: Bearer <Zabbix API token>`, отправляется с каждым запросом

Каждый пользователь должен настроить собственный API-токен.

Чтобы проверить настройку, отправьте один запрос к серверу MCP:

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

Код состояния HTTP 200 и список доступных инструментов в ответе подтверждают, что сервер MCP доступен и принимает токен; список показывает, какие инструменты предоставляются. Другие коды состояния указывают на следующие проблемы:

- 400 - заголовок `Accept` не содержит оба типа мультимедиа, требуемых транспортом;
- 401 - отсутствует заголовок `Authorization` с токеном;
- 403 - адрес клиента не указан в параметре `AllowedIP`.

Подтверждение и доступ к инструментам Если MCP-клиент поддерживает получение данных через формы, каждое изменение должно быть подтверждено пользователем до его отправки в Zabbix. MCP-клиент показывает пользователю, что будет изменено, и изменение применяется только после подтверждения пользователя; если пользователь отклоняет или закрывает запрос, в Zabbix ничего не изменяется. Подтверждение действительно в течение пяти минут и только для точно показанного изменения; в течение этого времени подтверждение можно использовать повторно.

Если MCP-клиент не поддерживает получение данных через формы (включая клиенты, поддерживающие только получение данных через URL), изменения применяются немедленно, без подтверждения. В этом случае ограничьте инструменты записи и разрешения API Zabbix.

Подтверждение особенно важно, поскольку вывод инструментов может содержать ненадежный текст из отслеживаемых систем, который способен повлиять на агента.

Используйте AllowTool, DenyTool, AllowToolRegexp и DenyToolRegexp, чтобы управлять инструментами, доступными MCP-серверу. Запрещенные инструменты нельзя вызывать или просматривать, а связанные с ними ресурсы недоступны.

> **Warning:**
> Правила AllowTool, DenyTool, AllowToolRegexp и DenyToolRegexp ограничивают доступ к MCP, но не изменяют разрешения токена Zabbix. Если адрес веб-интерфейса доступен агенту с искусственным интеллектом, агент может попытаться отправить прямые вызовы API. Не предоставляйте агенту доступ к расположению веб-интерфейса Zabbix и конфигурации MCP-сервера, чтобы запросы обязательно проходили через MCP-сервер и его средства защиты.

Рекомендации по обеспечению безопасности Токены API

Аутентификация и авторизация полностью выполняются Zabbix. Клиент MCP отправляет токен API Zabbix с каждым запросом; MCP-сервер передаёт его без сохранения или записи в журнал, а Zabbix проверяет токен, применяет разрешения и записывает действия в журнал аудита.

Используйте отдельный токен с минимальными привилегиями и сроком действия для каждого агента и пользователя. Храните токены в безопасном месте и регулярно выполняйте их ротацию.

Шифрование TLS

Чтобы токены не передавались в открытом виде, установите `TLSAccept=cert` и укажите сертификат и ключ (`TLSCertFile` и `TLSKeyFile`) либо запустите MCP-сервер за доверенным обратным прокси, завершающим TLS, и установите `TLSAccept=unencrypted`. Нешифрованный HTTP допустим только в доверенной сети.

Ограничения доступа

Ограничьте параметр `AllowedIP` адресами доверенных клиентов MCP и используйте разрешения Zabbix, чтобы ограничить возможности токенов API.

MCP-сервер не ограничивает частоту запросов. Каждый запрос MCP обычно приводит к одному вызову API Zabbix, однако агент, работающий в цикле, всё равно может создавать значительную нагрузку.

Защита конфиденциальных данных

Ключи элементов данных могут содержать учетные данные (например, в элементах данных SSH, JMX, ODBC и HTTP agent), значения макросов могут содержать пароли и строки сообществ, интерфейсы раскрывают внутренние адреса, а значения элементов данных журнала или текста могут содержать любые данные. По этой причине инструменты `item_get`, `lld_get`, `interface_get`, `macro_get` и `history_get` по умолчанию запрещены в поставляемом файле конфигурации.

Чтобы избежать раскрытия конфиденциальных значений макросов поставщику LLM подключенного агента, используйте секретные макросы; их значения никогда не возвращаются API.

Инструменты и ресурсы

Обзор

MCP-сервер Zabbix предоставляет клиентам MCP функциональность Zabbix в виде *инструментов* и *ресурсов*.

Названия инструментов описывают выполняемое действие. Если задан параметр ToolPrefix, этот префикс добавляется к названию каждого инструмента; правила доступа к инструментам сопоставляются с указанными ниже названиями без префикса.

Инструменты чтения

Инструмент	Метод Zabbix API	Аргументы	Сортировка по	Ресурс
hostgroup_get	hostgroup.get	groupids, names, search, limit	name	zabbix://hostgroup/<i
templategroup_get	templategroup.get	groupids, names, search, limit	name	zabbix://templategrou
host_get	host.get	hostids, groupids, search, limit	name	zabbix://host/<id>
template_get	template.get	templateids, groupids, search, limit	name	zabbix://template/<id
item_get	item.get	hostids, itemids, search, limit	name	zabbix://item/<id>
lld_get	discoveryrule.get	hostids, ruleids, limit	name	-
interface_get	hostinterface.get	hostids, interfaceids, limit	идентификатор интерфейса	-
macro_get	usermacro.get	hostids или global, macro_names, limit	макрос	-
problem_get	problem.get	eventids, groupids, hostids, severities, acknowledged, suppressed, recent, limit	идентификатор события, по убыванию	-
maintenance_get	maintenance.get	maintenanceids, groupids, hostids, limit	name	-
history_get	history.get	itemids, time_from, time_till, type, limit	время, по убыванию	-
trigger_get	trigger.get	triggerids, hostids, search, limit	описание	zabbix://trigger/<id>

Обратите внимание:

- host_get и template_get также возвращают триггеры каждого объекта (ограниченный набор полей), теги и связанные шаблоны. Элементы данных, правила обнаружения, интерфейсы и макросы возвращаются соответствующими инструментами.
- history_get требует диапазон времени не более 7 дней, начинающийся не ранее чем 7 дней назад. Аргумент type должен иметь одно из значений float, unsigned, character, text, log или json и соответствовать типу значения элементов данных; если тип указан неправильно, возвращается пустой результат, который невозможно отличить от элемента данных без данных. Историю двоичных элементов данных (тип значения 5) нельзя получить через MCP-сервер.
- Инструменты чтения запрашивают фиксированный список полей; все поля (output: extend) никогда не запрашиваются.
- Результаты инструментов, у которых есть ресурс, содержат поле uri, чтобы агент мог перейти от списка к отдельной сущности.

Формат ответа

Все инструменты чтения возвращают ответ в одном и том же формате:

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Ответ ограничивается аргументом limit (по умолчанию: 100, максимум: 1000) и сериализованным размером ответа (MaxResponseSize, по умолчанию: 64K). Формирование ответа прекращается при достижении любого из этих ограничений. Ответ усекается только между результатами, но не внутри результата; в усечённом ответе указывается, какое ограничение было достигнуто. Для каждого инструмента задан фиксированный порядок сортировки, поэтому усечённый ответ всегда содержит начало полного набора результатов. Постраничная выдача не поддерживается; чтобы получить другие результаты, сузьте запрос.

Инструменты для работы с проблемами

Все инструменты для работы с проблемами вызывают event.acknowledge с соответствующими флагами действий. Аргумент eventids является обязательным; идентификаторы событий можно получить с помощью problem_get. С помощью любого инструмента для работы с проблемами можно добавить сообщение (комментарий).

Аргумент message является необязательным, за исключением problem_annotate, где он обязателен; если он указан, сообщение записывается в рамках того же вызова API Zabbix.

Tool	Effect	Additional arguments
problem_acknowledge	Подтвердить проблему.	-
problem_unacknowledge	Удалить подтверждение.	-
problem_close	Закрыть проблему вручную (только для триггеров, допускающих ручное закрытие).	-
problem_annotate	Записать сообщение.	message (обязательный)
problem_change_severity	Изменить уровень серьезности проблемы.	severity (0-5)
problem_suppress	Подавить проблему до указанного времени или на неопределенный срок.	suppress_until
problem_unsuppress	Удалить подавление.	-

Проблемы нельзя создавать или удалять; все инструменты для работы с проблемами обновляют существующие проблемы.

Инструменты обслуживания

Инструмент	Метод Zabbix API	Аргументы
maintenance_create	maintenance.create	name, groupids и/или hostids, active_since, active_till, timeperiods, description, tags, tags_evaltype
maintenance_update	maintenance.update	maintenanceid и любые из указанных выше аргументов. Аргументы-списки заменяют весь набор значений; пустой список очищает набор.
maintenance_delete	maintenance.delete	maintenanceids

При обновлении периода обслуживания аргументы-списки (например, hostids, groupids и timeperiods) заменяют существующие значения, а не добавляются к ним. Например, если период обслуживания охватывает узлы сети A и B, а вызывается maintenance_update с hostids, содержащим только узел сети C, период обслуживания будет охватывать только узел сети C.

maintenance_create — единственная операция, которая не является идемпотентной, а maintenance_delete — единственная операция, которую невозможно отменить. Чтобы запретить их использование, запретите эти инструменты с помощью параметра DenyTool.

Поведение инструментов записи

Инструменты для работы с проблемами и обслуживаниями являются *инструментами записи*. Они работают следующим образом:

- Аргументы проверяются до любого вызова API Zabbix; если проверка не пройдена, в ошибке указывается невыполненное ограничение.
- Если клиент поддерживает эту возможность, изменение применяется только после его одобрения пользователем.
- Изменение выполняется Zabbix с использованием токена вызывающей стороны. Zabbix проверяет права доступа и записывает изменение в журнал аудита с указанием владельца токена.
- Ошибки Zabbix, включая ошибки доступа, возвращаются без изменений в виде объекта ошибки с полями code, message и data.
- Инструменты записи напрямую возвращают результат Zabbix; ограничения на размер ответа инструментов чтения не применяются.
- Вызовы записи никогда не повторяются MCP-сервером.

Ресурсы

Ресурсы используют схему URI zabbix://<kind>/<id>. Только zabbix://about включается в ответ resources/list; идентификаторы сущностей нельзя перечислить, их получают из результатов работы инструментов. В ответе resources/templates/list перечислены виды сущностей.

URI	Содержимое
zabbix://about	URL API, URL веб-интерфейса, версия Zabbix, идентификатор пользователя, которому соответствует токен вызывающей стороны, и список доступных инструментов.

URI	Содержимое
zabbix://host/<hostid>	Запись узла сети без элементов данных, триггеров, интерфейсов и макросов.
zabbix://hostgroup/<groupid>	Запись группы узлов сети без списка участников.
zabbix://template/<templateid>	Запись шаблона без элементов данных, триггеров и правил обнаружения.
zabbix://templategroup/<groupid>	Запись группы шаблонов без списка участников.
zabbix://item/<itemid>	Определение элемента данных без текущего значения.
zabbix://trigger/<triggerid>	Определение триггера без узлов сети, зависимостей, функций и тегов.

Ресурс, для которого соответствующий инструмент запрещен, не существует. Несуществующая сущность и сущность, к которой владелец токена не имеет доступа, в обоих случаях обозначаются как *ресурс не найден*; эти случаи намеренно неразличимы. На ресурс zabbix://about правила доступа к инструментам не распространяются.

Параметры конфигурации: Zabbix MCP server

Обзор

Сервер Zabbix MCP — это процесс, который позволяет агентам ИИ работать с Zabbix с использованием протокола контекста модели (MCP).

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации сервера Zabbix MCP (zabbix_mcp_server.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы просмотреть подробные сведения.

Параметр	Описание
AllowedIP	Список разделенных запятыми IP-адресов, при необходимости в нотации CIDR, или DNS-имен клиентов MCP.
AllowTool	Правило с подстановочными знаками, разрешающее инструмент.
AllowToolRegexp	Правило с регулярным выражением, разрешающее инструменты.
DebugLevel	Уровень отладки.
DenyTool	Правило с подстановочными знаками, запрещающее инструмент.
DenyToolRegexp	Правило с регулярным выражением, запрещающее инструменты.
FrontendURL	Конечная точка API Zabbix.
IgnoreURLCertErrors	Определяет обработку ошибок проверки сертификата TLS при доступе к API Zabbix.
ListenIP	IP-адрес, на котором выполняется прослушивание.
ListenPort	Порт, на котором выполняется прослушивание.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogType	Тип вывода журнала.
MaxResponseSize	Максимальный размер ответа, возвращаемого инструментом чтения.
PidFile	Имя PID-файла.
Timeout	Максимальное время (в секундах) ожидания ответа от API Zabbix.
TLSAccept	Тип входящих соединений, которые следует принимать.
TLSCertFile	Полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ сервера.
ToolPrefix	Префикс, добавляемый к именам всех инструментов.

Все параметры являются необязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для процесса, а не значения в поставляемых файлах конфигурации;
- Значения поддерживают переменные окружения в формате ${NAME};
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметрах

AllowedIP

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён клиентов MCP. Входящие подключения будут приниматься только от узлов сети, перечисленных здесь.

Пример:

`AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com`

Обязательный: да

AllowTool

Правило с подстановочными символами, разрешающее использование инструмента. Параметр можно указывать несколько раз.
 Шаблоны с подстановочными символами используют синтаксис ключа элемента данных, где * соответствует любым символам. Правило сопоставляется с простым именем инструмента и игнорирует ToolPrefix.
 Все правила `AllowTool`, `DenyTool`, `AllowToolRegexp` и `DenyToolRegexp` образуют единый список в порядке их указания; решение принимает первое совпавшее правило. Инструмент, которому не соответствует ни одно правило, разрешен.

Пример:

`AllowTool=host_get`

AllowToolRegexp

Правило регулярного выражения, разрешающее использование инструментов. Параметр можно указывать несколько раз.
 В регулярных выражениях используется синтаксис RE2; если выражение не привязано к началу или концу строки, оно сопоставляется с любой частью обычного имени инструмента.

Пример:

`AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$`

DebugLevel

Укажите уровень отладки: - *0* - основная информация о запуске и остановке процессов Zabbix; - *1* - критическая информация; - *2* - информация об ошибках; - *3* - предупреждения; - *4* - отладка (выводится большой объем информации); - *5* - расширенная отладка (выводится еще больше информации).

Значение по умолчанию: 3
 Диапазон: 0-5

DenyTool

Правило с подстановочными знаками, запрещающее использование инструмента. Параметр можно указывать несколько раз.
 Запрещенный инструмент не включается в список инструментов и не может быть вызван. Подробную информацию см. в разделе AllowTool.

Пример:

`DenyTool=maintenance_delete`

DenyToolRegexp

Правило регулярного выражения, запрещающее использование инструментов. Параметр можно указывать несколько раз.
 Подробнее см. в разделе AllowToolRegexp.

Пример:

`DenyToolRegexp=^problem_(close|suppress)$`

FrontendURL

Конечная точка API Zabbix: URL веб-интерфейса Zabbix, оканчивающийся на `api_jsonrpc.php`.

Пример:

`FrontendURL=https://zabbix.example.com/api_jsonrpc.php`

Обязательный параметр: да

IgnoreURLCertErrors

Определяет обработку ошибок проверки сертификата TLS при обращении к API Zabbix:
 false - не игнорировать ошибки сертификата;
 true - игнорировать ошибки сертификата (используйте только в среде разработки).

Значение по умолчанию: `false`

ListenIP

IP-адрес для прослушивания.

По умолчанию: `0.0.0.0`

ListenPort

Порт для прослушивания. Путь конечной точки MCP: `/mcp`.

Значение по умолчанию: 8443
 Диапазон: 1024–32767

LogFile

Имя файла журнала.

По умолчанию: `/tmp/zabbix_mcp_server.log`

Обязательный: нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 - отключить автоматическую ротацию журналов.

Значение по умолчанию: 1
 Диапазон: 0-1024

LogType

Тип вывода журнала: - *file* — записывать журнал в файл, указанный параметром LogFile; - *system* — записывать журнал в syslog; - *console* — записывать журнал в стандартный вывод.

Значение по умолчанию: `file`

MaxResponseSize

Максимальный размер ответа, возвращаемого инструментом чтения. Ответ прекращается при достижении первого из ограничений: этого значения или аргумента `limit` инструмента.
 Поддерживаются суффиксы единиц измерения.

Значение по умолчанию: 64K
 Диапазон: 16K-256K

PidFile

Имя PID-файла.

По умолчанию: `/tmp/zabbix_mcp_server.pid`

Тайм-аут

Максимальное время (в секундах) ожидания ответа от API Zabbix.

По умолчанию: 10
 Диапазон: 3-30

TLSAccept

Какие входящие подключения принимать: - *unencrypted* - принимать подключения без шифрования (по умолчанию); - *cert* - принимать подключения с TLS и сертификатом.

По умолчанию: `unencrypted`

TLSCertFile

Полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов.

Обязательный параметр: да, если `TLSAccept` установлен в значение `cert`; в противном случае — нет

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ сервера.

Обязательно: да, если для `TLSAccept` задано значение `cert`; в противном случае — нет

ToolPrefix

Префикс, добавляемый к именам всех инструментов, например для предотвращения конфликтов имен с инструментами других MCP-серверов. Например, при использовании `ToolPrefix=zbx_` инструмент `host_get` предоставляется как `zbx_host_get`.
 Допустимые символы: буквы, цифры, символ подчеркивания (`_`), дефис (`-`) и точка (`.`).
 Правила доступа к инструментам соответствуют именам инструментов без префикса.

Пример:

```
ToolPrefix=zbx_
```

Адаптер моста

Адаптер моста Zabbix позволяет вашему серверу Zabbix обмениваться данными с мобильным приложением Zabbix.

Мобильное приложение Zabbix не подключается к вашему серверу Zabbix напрямую. Вместо этого два компонента передают данные между ними:

- Zabbix Bridge — централизованно размещённая служба, которая обменивается данными с мобильным приложением Zabbix на вашем мобильном устройстве.
- Адаптер моста Zabbix — компонент, который необходимо установить и запустить. Он подключает ваш сервер Zabbix к Zabbix Bridge.

Адаптер моста можно установить в той же системе, что и ваш сервер Zabbix, или в другой системе, используя пакеты.

Чтобы настроить адаптер моста Zabbix, измените его файл конфигурации (путь по умолчанию: `/etc/zabbix/zabbix_bridge_adap`

Чтобы обеспечить безопасность адаптера моста Zabbix, см. раздел Безопасный мобильный доступ.

Поток данных При использовании мобильного приложения Zabbix данные передаются следующим образом:

1. Вы выполняете действие в мобильном приложении Zabbix, после чего оно отправляет запрос в Zabbix Bridge.
2. Zabbix Bridge передаёт запрос адаптеру Zabbix bridge в виде запроса JSON-RPC.
3. Адаптер Zabbix bridge проверяет, разрешён ли запрошенный метод Zabbix API.
   Если метод не разрешён, адаптер отклоняет запрос.
   Если метод разрешён, адаптер отправляет запрос в Zabbix API.
4. Zabbix API обрабатывает запрос и возвращает ответ адаптеру Zabbix bridge.
5. Адаптер Zabbix bridge возвращает ответ в Zabbix Bridge.
6. Zabbix Bridge передаёт ответ мобильному приложению Zabbix, которое отображает данные на вашем мобильном устройстве.

Параметры конфигурации: адаптер моста Zabbix

Обзор

Параметры конфигурации позволяют настраивать поведение адаптера моста Zabbix.

Укажите значения параметров в файле конфигурации (`zabbix_bridge_adapter.conf`). Все параметры являются **необязательными**, если явно не указано, что параметр является обязательным.

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его имя)
- Подробные сведения о параметрах

Параметр	Описание
AllowedAPIMethods	Укажите, какие методы API Zabbix можно вызывать через адаптер моста Zabbix.
DataDir	Укажите каталог, в котором создаются и хранятся файл состояния адаптера моста и ключи шифрования.
DebugLevel	Установите уровень отладки.
Include	Укажите каталог или отдельные файлы, которые необходимо включить в файл конфигурации.
ListenIP	Укажите IP-адрес, на котором адаптер моста принимает входящие HTTP(S)-подключения от сервера Zabbix.
ListenHealthPort	Укажите порт, на котором адаптер моста принимает входящие HTTP-подключения для проверки работоспособности.
ListenPort	Укажите порт, на котором адаптер моста принимает входящие HTTP(S)-подключения от сервера Zabbix.
LogFile	Укажите полный путь к файлу журнала адаптера моста.
LogFileSize	Установите максимальный размер файла журнала (в МБ).
LogType	Укажите тип вывода журнала.
PidFile	Укажите файл PID.
TLSConfig.AllowedCN	Укажите разделенный запятыми список разрешенных общих имен клиентов (mTLS только).

Параметр	Описание
TLSConfig.CAFile	Укажите полный путь к файлу, содержащему сертификат(ы) корневого центра сертификации, используемые для проверки сертификатов клиентов в режиме mTLS.
TLSConfig.CertFile	Укажите полный путь к файлу, содержащему сертификат или цепочку сертификатов.
TLSConfig.KeyFile	Укажите полный путь к файлу, содержащему закрытый ключ.
TLSConfig.Mode	Установите режим TLS для входящих подключений на ListenPort.
ServerId	Укажите уникальный идентификатор сервера Zabbix. **Обязательный параметр**.
ZabbixAPITLSCAFile	Укажите полный путь к файлу, содержащему сертификат(ы) корневого центра сертификации, используемые адаптером моста для проверки сертификата API Zabbix.
ZabbixAPITLSConnect	Установите режим TLS для исходящих подключений к API Zabbix.
ZabbixAPIURL	Укажите URL конечной точки API Zabbix. **Обязательный параметр**.
ZabbixBridgeURL	Укажите URL моста Zabbix для получения конфигурации моста Zabbix (получается при запуске адаптера моста).

Обратите внимание:

- Значения по умолчанию отражают значения по умолчанию для процесса, а не значения в поставляемых файлах конфигурации.
- Значения поддерживают переменные окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с #, поддерживаются только в начале строки.

Сведения о параметрах

AllowedAPIMethods

Укажите, какие методы API Zabbix можно вызывать через адаптер моста Zabbix.
 Для нескольких методов повторите параметр, указывая по одному методу в каждой строке.

Каждое значение параметра должно быть либо точным именем метода (например, problem.get), либо шаблоном с подстановочным символом, который заменяет только полное имя API (например, *.get) или только полное имя метода (например, problem.*).

Любое другое значение является недопустимым и препятствует запуску адаптера моста Zabbix. К таким значениям относятся одиночный подстановочный символ (*), шаблон с частичным подстановочным символом (host* или host*.*) и пустое значение (AllowedAPIMethods=).

Пример:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

Минимальный набор методов, необходимый мобильному приложению Zabbix:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

По умолчанию: не задано (все методы API Zabbix запрещены)

DataDir

Укажите каталог, в котором создаются и хранятся файл состояния адаптера моста и ключи шифрования.
 Каталог должен иметь права доступа 0700, а все хранящиеся в нем файлы — права доступа 0600.

Значение по умолчанию: /var/lib/zabbix-bridge-adapter

DebugLevel

Установите уровень отладки:

- 0 — основная информация о запуске и остановке процессов Zabbix
- 1 — критическая информация
- 2 — информация об ошибках
- 3 — предупреждения
- 4 — отладка (создает большой объем информации)
- 5 — расширенная отладка (создает еще больший объем информации)

Значение по умолчанию: 3
 Диапазон: 0-5

Подключение

Укажите каталог или отдельные файлы, которые необходимо подключить в файл конфигурации.
 При установке Zabbix этому параметру присваивается значение `/usr/local/etc`, если во время компиляции не указано другое значение.

Общие правила для параметра `Include`:

- Путь может быть относительным к расположению файла `zabbix_bridge_adapter.conf`.
- В одном файле конфигурации можно указать несколько параметров `Include`.
- Zabbix обрабатывает параметр `Include` сразу после его чтения, поэтому рекомендуется размещать его в конце файла конфигурации.

Правила для подключаемых файлов:

- Каждый файл должен быть доступен для чтения.
- Каждый файл должен иметь корректный синтаксис. В противном случае компонент не запустится.

Правила подключения каталогов или шаблонов файлов:

- Не следует рассчитывать на определённый порядок подключения (например, файлы не подключаются в алфавитном порядке).
- Не определяйте один и тот же параметр более чем в одном подключаемом файле (например, чтобы переопределить общую настройку конкретной), поскольку невозможно предсказать, какое значение будет использовано.
- Обращайте внимание на резервные копии, которые некоторые текстовые редакторы создают автоматически (например, при редактировании `include/my_specific.conf` может быть создан файл `include/my_specific.conf.BAK`), поскольку будут подключены оба файла.

Чтобы подключить из каталога только нужные файлы, используйте подстановочный символ (`*`) для сопоставления с шаблоном.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Укажите IP-адрес прослушивания адаптера моста для входящих HTTP(S)-подключений от сервера Zabbix.
 Если адрес не указан, адаптер моста прослушивает все IP-адреса.

ListenHealthPort

Укажите порт прослушивания адаптера моста для входящих HTTP-подключений проверки состояния.
 Обслуживает конечные точки `/health` и `/ready`.
 Если параметр не указан, служба проверки состояния не запускается.

Значение по умолчанию: 8081
 Диапазон: 1024-32767

ListenPort

Укажите порт прослушивания адаптера моста для входящих HTTP(S)-подключений от сервера Zabbix.

Значение по умолчанию: 10054
 Диапазон: 1024–32767

LogFile

Укажите полный путь к файлу журнала адаптера моста.

Обязательно: **Да**, если LogType имеет значение `file`

По умолчанию: `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Задайте максимальный размер файла журнала (в МБ).
 0 - отключить автоматическую ротацию журналов.

По умолчанию: 1
 Диапазон: 0-1024

LogType

Укажите тип вывода журнала:

- `file` - записывать журнал в файл, указанный параметром `LogFile`.
- `system` - записывать журнал в системный журнал.
- `console` - записывать журнал в стандартный вывод.

Значение по умолчанию: `file`

PidFile

Укажите PID-файл.

Значение по умолчанию: `/tmp/zabbix_bridge_adapter.pid`

ServerId

Укажите уникальный идентификатор сервера Zabbix.
 Доступен в разделе *Отчёты* > *Сведения о системе* веб-интерфейса Zabbix.

Обязательный параметр: **Да**

TLSConfig.AllowedCN

Укажите разделенный запятыми список разрешенных общих имен клиентов (только для `mTLS`).
 Если параметр не указан, адаптер моста не проверяет общее имя клиента.

TLSConfig.CAFile

Укажите полный путь к файлу, содержащему сертификат(ы) корневого центра сертификации, используемые для проверки сертификатов клиентов в режиме `mTLS`.

TLSConfig.CertFile

Укажите полный путь к файлу, содержащему сертификат или цепочку сертификатов.

TLSConfig.KeyFile

Укажите полный путь к файлу, содержащему закрытый ключ.

TLSConfig.Mode

Задайте режим TLS для входящих подключений на `ListenPort`:

- `none` - принимать подключения без шифрования.
- `tls` - стандартный TLS с минимальной версией v1.2.
- `mtls` - расширяет TLS взаимной аутентификацией.

Значение по умолчанию: `none`

ZabbixAPITLSCAFile

Укажите полный путь к файлу, содержащему сертификат(ы) корневого центра сертификации, используемые адаптером моста для проверки сертификата API Zabbix.

Обязательный параметр: **Да**, если `ZabbixAPITLSConnect` имеет значение `verify_ca` или `verify_full`.

ZabbixAPITLSConnect

Задайте режим TLS для исходящих подключений к API Zabbix:

- `disabled` (или пустое значение) — использовать обычный HTTP; `ZabbixAPIURL` должен использовать `http://`.
- `required` — использовать HTTPS без проверки сертификата (только для тестирования).
- `verify_ca` — использовать HTTPS и проверять сертификат API Zabbix.
- `verify_full` — использовать HTTPS и проверять сертификат API Zabbix и имя узла.

ZabbixAPIURL

Укажите URL конечной точки API Zabbix.

Обязательный параметр: **Да**

Пример:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
```

ZabbixBridgeURL

Укажите URL моста Zabbix для получения конфигурации моста Zabbix (получается при запуске адаптера моста).

Значение по умолчанию: ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration

Переменные среды

Обзор Переменные окружения позволяют настраивать компоненты Zabbix без жесткого задания значений в файлах конфигурации. Это упрощает управление конфигурациями в динамических средах, таких как Docker, где переменные можно передавать во время выполнения, чтобы адаптировать систему к разным сценариям.

В простейшем случае, задав значение параметра конфигурации Zabbix server DebugLevel через переменную окружения, вы можете использовать ее для настройки сервера при запуске:

```
###            Zabbix server:
DebugLevel=${NEW_DEBUG_LEVEL}

###     Zabbix server:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Переменные окружения поддерживаются следующими компонентами Zabbix:

- Сервер
- Прокси
- Агент (UNIX или Windows)
- Агент 2 (UNIX или Windows), включая плагины
- Веб-сервис
- Zabbix sender (при использовании опции -c, --config)

Важные примечания

- Когда параметр конфигурации задан через переменную среды, которая не указана при запуске компонента, используется значение параметра по умолчанию.
- При использовании команд во время выполнения (например, для увеличения уровня журнала агента) необходимо указывать все ранее использовавшиеся переменные среды. Это связано с тем, что компоненты Zabbix используют свой файл конфигурации для выполнения команд во время выполнения; если переменные среды опущены, будут использованы значения параметров конфигурации по умолчанию. См. Примеры.
- Команда во время выполнения `userparameter_reload` runtime command не поддерживает повторную загрузку переменных среды. Во время перезагрузки переменные игнорируются, и перезагружаются только параметры с обычными значениями.
- Текущие переменные среды процесса, которые использовались в файлах конфигурации, очищаются после запуска компонента Zabbix. Это гарантирует, что дочерние процессы (например, удаленные скрипты, выполняемые Zabbix) не смогут получить доступ к этим переменным. Однако обратите внимание, что начальные переменные процесса по-прежнему можно получить (например, через файл `/proc/<PID>/environ`).

Синтаксис Переменные среды должны использовать следующий синтаксис: ${буквенно-цифровые_символы/подчеркивания}.

Имя переменной может содержать только буквы (a-z, A-Z), символы подчеркивания (_), и цифры (0-9), и не должно начинаться с цифры.

Переменные, которые не соответствуют требуемому синтаксису или объединены с обычным значением, будут рассматриваться как обычные значения, что может привести к ошибкам.

Правильный синтаксис переменной:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Неправильный синтаксис переменной:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
В Windows имена переменных среды не чувствительны к регистру.

Примеры Следующие примеры показывают, как настроить и использовать переменные среды с компонентами Zabbix.

Пример 1: Настройка и тестирование агента Zabbix

1. Задайте переменные среды в файле конфигурации агента:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Протестируйте файл конфигурации:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. Запустите агент с переменными среды:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

При использовании команд во время выполнения (например, для повышения уровня журнала агента) необходимо указать все ранее использованные переменные среды:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Это связано с тем, что агент использует свой файл конфигурации для выполнения команд во время выполнения; если переменные среды не указаны, будут использоваться значения параметров конфигурации по умолчанию.

Кроме того, после задания переменных среды в файле конфигурации агента можно сделать их доступными для процессов (например, с помощью команды `export`). Это снижает риск неожиданного поведения из-за отсутствующих или заданных неверно переменных.

```
export ZBX_HOSTNAME="New Zabbix agent"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

Пример 2: Настройка агента Zabbix для контейнера

Если вы создаете и настраиваете собственный образ для компонентов Zabbix (например, агента Zabbix), можно определить параметры конфигурации с помощью переменных среды, а затем запустить контейнер с этими переменными.

1. При подготовке образа задайте переменные среды в файле конфигурации агента:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. После сборки образа контейнера запустите контейнер агента (например, Docker) с переменными среды:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. При использовании команд, выполняемых во время работы (например, для повышения уровня журнала агента) войдите в оболочку контейнера и выполните команду, выполняемую во время работы:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
Команда, выполняемая во время работы `userparameter_reload`, не поддерживает перезагрузку переменных среды. Во время перезагрузки переменные игнорируются, а перезагружаются только параметры с обычными значениями.

Конфигурация

Обзор Настройка в Zabbix включает указание того, какие узлы сети и системы следует мониторить, определение того, какие данные собирать, а также настройку того, как доставляются уведомления при возникновении проблем.

Используйте боковую панель для перехода к соответствующим аспектам настройки.

Основные задачи настройки

- Настроить новый узел сети или группу узлов сети → Hosts and host groups
- Определить, какие данные собирать с узлов сети → Items
- Создать правила для обнаружения проблем → Problem detection with triggers
- Настроить способ доставки уведомлений → Notifications upon events
- Отображать данные мониторинга с помощью панелей, графиков и карт → Visualization
- Повторно использовать конфигурации для нескольких узлов сети → Templates and template groups, Templates out of the box
- Управлять доступом пользователей и правами доступа → Users and user groups

Лучшие практики конфигурации Выбор конфигурации влияет на производительность и удобство сопровождения. Хотя каждая установка Zabbix отличается, эти лучшие практики можно использовать как основу и при необходимости адаптировать.

Отдельный узел сети для каждого экземпляра мониторинга

Узел сети в Zabbix — это логическая сущность, а не физическая машина или устройство. Создавайте отдельный узел сети для каждого независимого экземпляра, например виртуальной машины, базы данных, контейнера или сетевого коммутатора.

Такой подход обеспечивает:

1. Более чистые данные мониторинга с отдельными элементами данных, триггерами и уведомлениями о событиях для каждого узла сети.
2. Точный контроль доступа пользователей через роли пользователей, соответствующий принципу наименьших привилегий.

Используйте шаблоны и массовое обновление для похожих узлов сети

Для похожих узлов сети, таких как *Network switch 1* и *Network switch 2*, вы можете создать шаблон с необходимыми метриками (*Network switch template*) и объединить узлы сети в группу узлов сети (*Network switches*).

При добавлении новых коммутаторов вы можете использовать массовое обновление в *Data Collection* > *Hosts*, чтобы связать шаблон вместо ручного клонирования узлов сети.

Настройка master-элемента и зависимых элементов для минимизации запросов

Чтобы минимизировать количество запросов к целевому объекту, настройте master-элемент для сбора данных одним запросом, а затем используйте зависимые элементы с предварительной обработкой для извлечения конкретных значений.

Например, master-элемент может собирать ответ JSON или XML с несколькими метриками либо выполнять запрос к базе данных, возвращающий несколько столбцов (например, количество открытых соединений, прерванных соединений, максимально допустимое число одновременных соединений и общее накопленное число соединений с момента запуска), а зависимые элементы будут разбирать и сохранять каждое значение отдельно.

Отбрасывайте историю master-элемента после сбора и сохраняйте только данные зависимых элементов.

Развертывание прокси Zabbix при необходимости

Если все узлы сети находятся в той же локальной сети, что и сервер Zabbix, и проблем с масштабируемостью или производительностью нет, прокси может не потребоваться. В более крупных или сложных средах прокси обеспечивает более равномерное распределение нагрузки.

Добавьте прокси Zabbix в следующих случаях:

1. Мониторинг нескольких узлов сети с использованием различных методов сбора метрик за межсетевым экраном. Прокси собирает данные с узлов сети и передает их серверу Zabbix, уменьшая количество открытых портов межсетевого экрана.
2. Мониторинг удаленных местоположений, филиалов или сетей. Прокси Zabbix в удаленных местоположениях продолжает сбор данных во время перебоев в сети и отправляет данные после восстановления соединения.

3. Управление крупномасштабными развертываниями. Если у вас много узлов сети и/или собирается большое количество значений в секунду, разверните прокси, чтобы снизить нагрузку на сервер и повысить производительность.

Настройка независимого получения секретных макросов

Вы можете использовать пользовательские макросы secret либо как секретный текст, либо как секретные макросы хранилища.

По умолчанию значения секретных макросов извлекаются сервером Zabbix и передаются прокси. Для повышения безопасности при использовании секретных макросов хранилища настройте значения макросов так, чтобы они извлекались независимо сервером Zabbix и прокси.

Настройка шаблона

Обзор

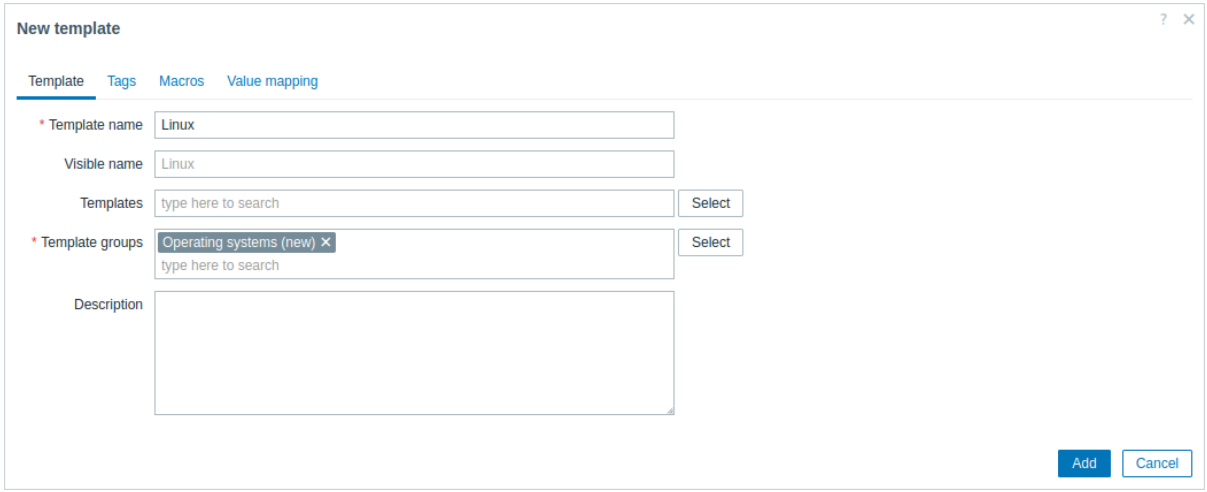
Настройка шаблона требует, чтобы сначала вы создали шаблон, задав его общие параметры, а затем добавили к нему сущности (элементы данных, триггеры, графики и т. д.).

Создание шаблона

Чтобы создать шаблон, выполните следующие действия:

1. Перейдите в раздел *Сбор данных* > *Шаблоны*.
2. Нажмите *Создать шаблон*.
3. Измените атрибуты шаблона.

Вкладка **Шаблон** содержит общие атрибуты шаблона.



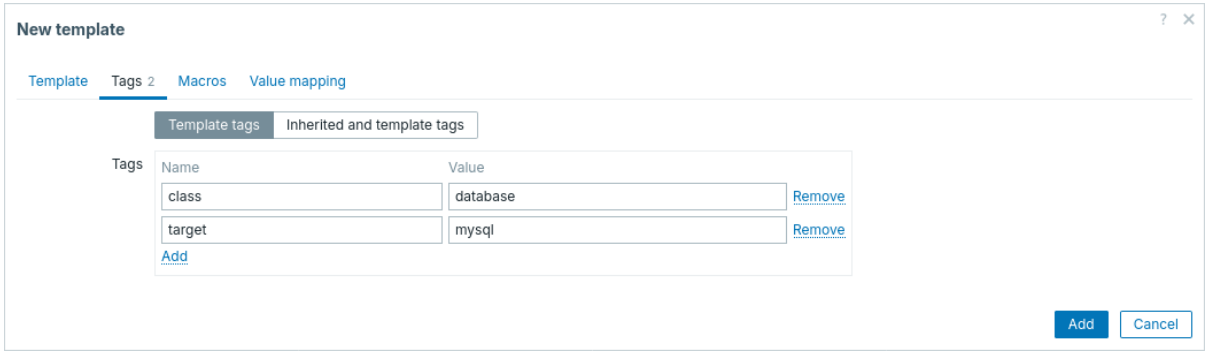
Все обязательные поля ввода отмечены красной звездочкой.

Атрибуты шаблона:

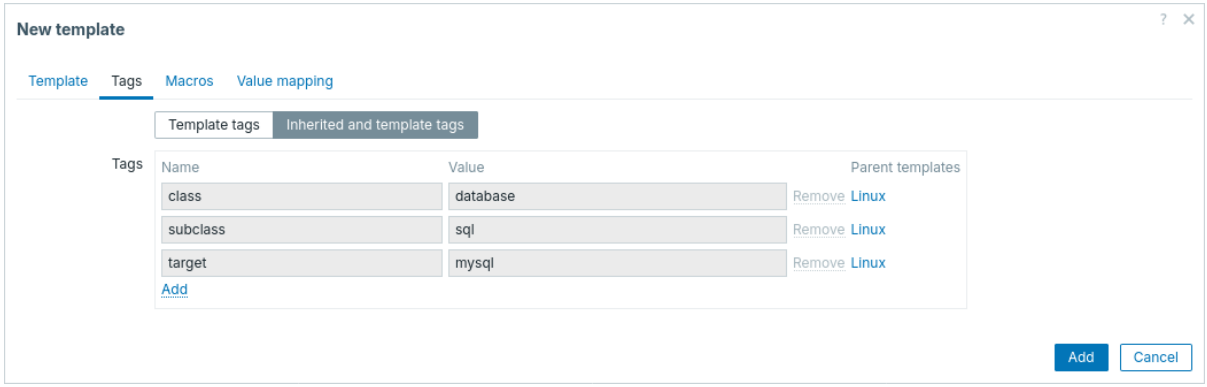
Параметр	Описание
Имя шаблона	Уникальное имя шаблона. Разрешены буквы и цифры, пробелы, точки, дефисы и символы подчеркивания. Пробелы в начале и конце не допускаются.
Отображаемое имя	Если указать это имя, оно будет отображаться в списках, на картах и т. д.

Параметр	Описание
Шаблоны	Свяжите с этим шаблоном один или несколько шаблонов. Все сущности (элементы данных, триггеры и т. д.) будут унаследованы от связанных шаблонов. Чтобы связать новый шаблон, введите его имя в поле *Шаблоны* — появится раскрывающийся список соответствующих шаблонов. Также можно нажать *Выбрать*, чтобы открыть всплывающее окно *Шаблоны*, которое изначально не содержит шаблонов. Чтобы заполнить список, введите имя группы шаблонов в поле *Группа шаблонов* (и выберите группу из предложенного списка) или нажмите кнопку *Выбрать* рядом с полем *Группа шаблонов*, чтобы открыть всплывающее окно *Группы шаблонов*. После выбора группы шаблонов всплывающее окно *Шаблоны* обновится и отобразит шаблоны, входящие в эту группу. Выбранные шаблоны будут связаны после сохранения или обновления конфигурации. Чтобы отменить связь с шаблоном, используйте один из двух вариантов в блоке *Шаблоны*: *Отменить связь* — отменить связь с шаблоном, сохранив его сущности (элементы данных, триггеры и т. д.); *Отменить связь и очистить* — отменить связь с шаблоном и удалить все его сущности (элементы данных, триггеры и т. д.).
Группы шаблонов	Группы шаблонов, к которым относится шаблон.
Описание	Описание шаблона.
Производитель и версия	Производитель и версия шаблона; отображаются только при обновлении существующих шаблонов (готовых шаблонов, предоставляемых Zabbix, импортированных шаблонов или шаблонов, измененных через API шаблонов), если конфигурация шаблона содержит такую информацию. Не могут быть изменены в веб-интерфейсе Zabbix. Для готовых шаблонов версия отображается следующим образом: основная версия Zabbix, разделитель (”-”), номер редакции (увеличивается с каждой новой версией шаблона и сбрасывается с каждой основной версией Zabbix). Например: 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Вкладка **Теги** позволяет определить теги на уровне шаблона. Все проблемы узлов сети, связанных с этим шаблоном, будут помечены указанными здесь значениями.



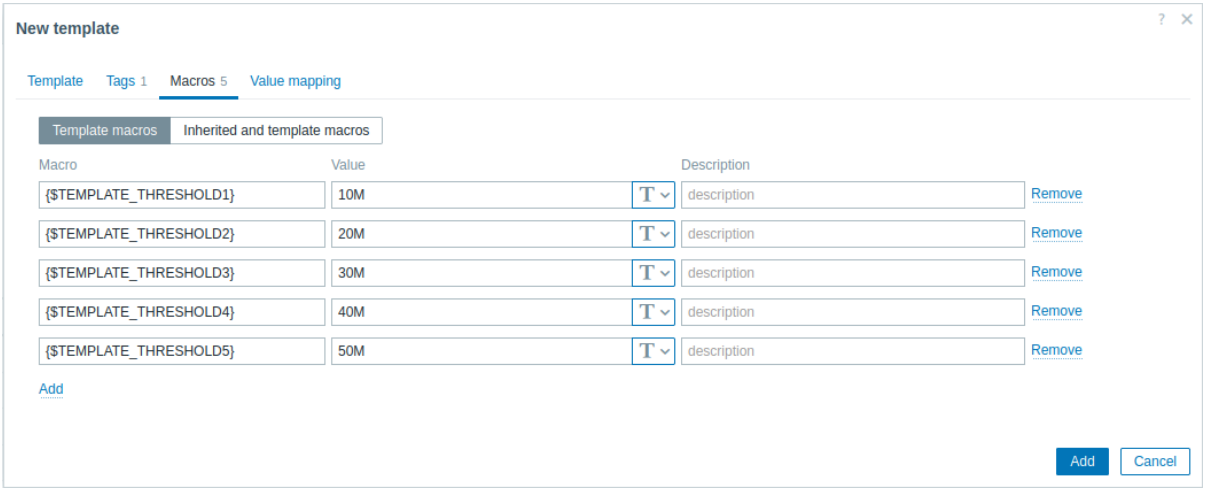
Если выбрать параметр *Унаследованные теги и теги шаблона*, можно также просматривать теги связанных шаблонов, которые будут унаследованы этим шаблоном.



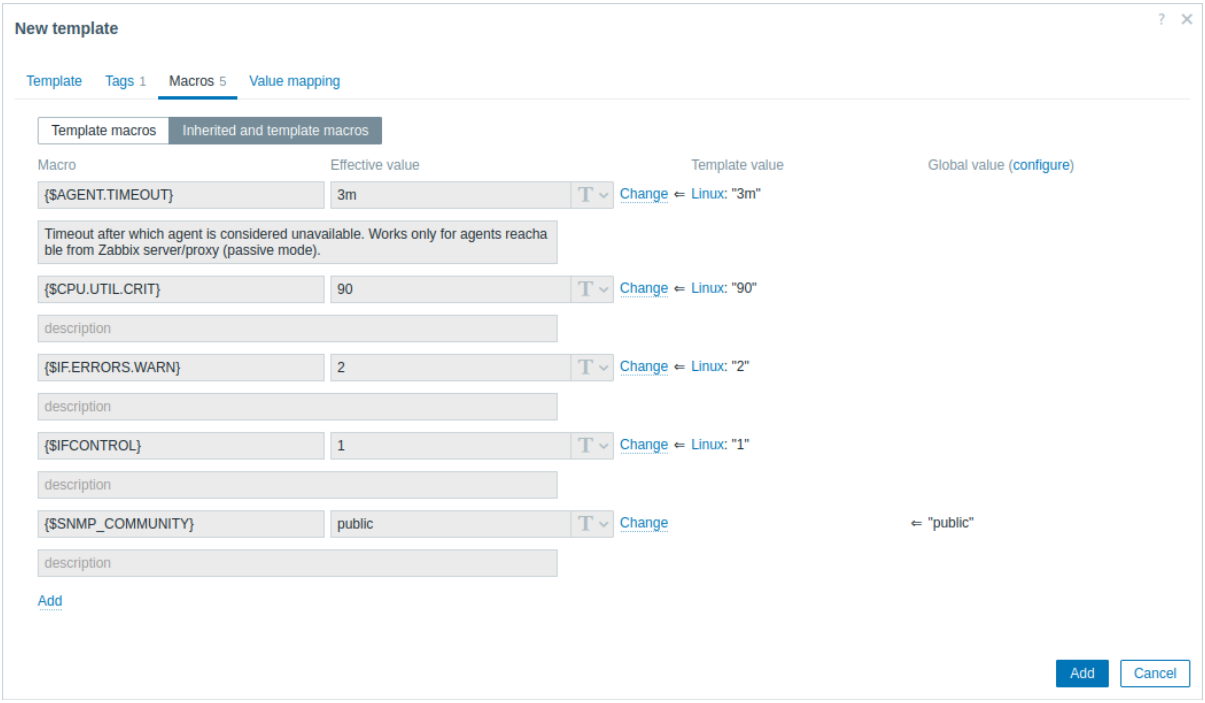
Для удобства предусмотрены ссылки на соответствующие шаблоны.

В тегах поддерживаются пользовательские макросы, макросы {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}.

Вкладка **Макросы** позволяет определить пользовательские макросы на уровне шаблона в виде пар «имя — значение». Обратите внимание, что значения макросов могут храниться как обычный текст, секретный текст или секрет Vault. Также можно добавить описание.



Если выбрать параметр *Унаследованные макросы и макросы шаблона*, можно также просматривать макросы связанных шаблонов и глобальные макросы, которые будут унаследованы этим шаблоном, а также значения, в которые будут разрешаться макросы.



Для удобства предусмотрены ссылки на соответствующие шаблоны, а также ссылка на конфигурацию глобальных макросов. Также можно изменить макрос связанного шаблона или глобальный макрос на уровне шаблона, фактически создав копию макроса в шаблоне.

Вкладка **Сопоставление значений** позволяет настроить удобное для пользователя представление данных элементов данных в сопоставлениях значений.

Кнопки:

Add	Добавить шаблон. Добавленный шаблон должен появиться в списке.
Update	Обновить свойства существующего шаблона.
Clone	Создать другой шаблон на основе свойств текущего шаблона. Он **включает** сущности (элементы данных, триггеры и т. д.), унаследованные от связанных шаблонов и непосредственно присоединенные к текущему шаблону, но **исключает** производителя и версию текущего шаблона, чтобы клонированный шаблон можно было отличить от исходного.
Delete	Удалить шаблон; сущности шаблона (элементы данных, триггеры и т. д.) останутся у связанных узлов сети.

Delete and clear	Удалить шаблон и все его сущности из связанных узлов сети.
Cancel	Отменить редактирование свойств шаблона.

Добавление элементов данных, триггеров, графиков

Attention:
Элементы данных сначала нужно добавить в шаблон. Триггеры и графики нельзя добавить без соответствующего элемента данных.

Вы можете добавить существующие элементы данных в шаблон или создать новые элементы данных.

Чтобы добавить элементы данных в шаблон:

1. Перейдите в *Сбор данных* > *Узлы сети* (или *Шаблоны*).
2. Нажмите *Элементы данных* в строке нужного узла сети/шаблона.
3. Отметьте флажками элементы данных, которые вы хотите добавить в шаблон.
4. Нажмите *Копировать* под списком элементов данных.
5. Выберите шаблон (или группу шаблонов), в который нужно скопировать элементы данных, и нажмите *Копировать*.
Все выбранные элементы данных должны быть скопированы в шаблон.

Добавление триггеров и графиков выполняется аналогичным образом (из списков триггеров и графиков соответственно), при этом также следует помнить, что их можно добавить только после добавления необходимых элементов данных.

Добавление дашбордов

Чтобы добавить дашборды в шаблон в *Сбор данных* > *Шаблоны*, выполните следующие действия:

1. Нажмите *Дашборды* в строке шаблона.
2. Настройте дашборд, следуя рекомендациям по настройке дашбордов.

Attention:
При настройке виджетов на дашборде шаблона (вместо глобального дашборда) параметры, связанные с узлом сети, недоступны, а некоторые параметры имеют другую подпись. Это связано с тем, что дашборды шаблона отображают данные только с узла сети, к которому привязан шаблон. Например, параметры *Группы узлов сети*, *Исключить группы узлов сети* и *Узлы сети* в виджете *Проблемы* недоступны, параметр *Группы узлов сети* в виджете *Доступность узлов сети* недоступен, а параметр *Показывать узлы сети в обслуживании* переименован в *Показывать данные в обслуживании* и т. д. Дополнительные сведения о доступности параметров в виджетах дашборда шаблона см. в описании конкретных параметров для каждого виджета дашборда.

Note:
Подробнее о доступе к дашбордам узлов сети, созданным из дашбордов шаблона, см. в разделе дашборды узлов сети.

Настройка правил низкоуровневого обнаружения

См. раздел низкоуровневое обнаружение руководства.

Добавление веб-сценариев

Чтобы добавить веб-сценарии в шаблон в *Сбор данных* > *Шаблоны*, выполните следующие действия:

1. Нажмите *Веб* в строке шаблона.
2. Настройте веб-сценарий обычным способом, как описано в настройке веб-сценариев.

Настройка группы шаблонов

Обзор

Группы шаблонов используются для логического объединения шаблонов и назначения им прав пользователей.

Каждый шаблон должен быть назначен как минимум в одну группу шаблонов. Шаблон может принадлежать нескольким группам шаблонов, и каждая группа шаблонов может содержать несколько шаблонов.

Обратите внимание, что в Zabbix все права основаны на группах: группах пользователей, группах узлов сети и группах шаблонов. Поэтому даже если одному пользователю нужен доступ к одному шаблону, он предоставляется путем

добавления пользователя в группу пользователей и назначения этой группе пользователей прав на группу шаблонов, содержащую этот шаблон.

Конфигурация

Attention:
Только пользователи с ролью Super admin могут создавать группы шаблонов.

В веб-интерфейсе Zabbix есть два способа создания группы шаблонов.

Вариант 1:

1. Перейдите в *Сбор данных* > *Группы шаблонов*.
2. Нажмите *Создать группу шаблонов* в правом верхнем углу экрана.
3. Введите имя группы в форме.



Вариант 2: при настройке шаблона введите несуществующее имя группы в поле *Группы шаблонов*.

После создания группы шаблонов можно нажать на имя шаблона в списке в разделе *Сбор данных* > *Группы шаблонов*, чтобы изменить имя группы, клонировать группу или удалить ее.

При удалении группы шаблонов удаляется только логическая группа, а не шаблоны в ней. Невозможно удалить группу шаблонов, если она является единственной группой для какого-либо существующего шаблона.

Создание подгрупп шаблонов

Подгруппа шаблонов (или вложенная группа шаблонов) является дочерней по отношению к родительской группе шаблонов, которая ее содержит.

Подгруппа создается с помощью прямой косой черты '/' в поле ввода имени группы, чтобы обозначить ее связь с родительской группой или группами. Например:

- ввод `Linux servers/Databases` создает подгруппу `Linux servers/Databases` в составе родительской группы `Linux servers`.
- ввод `Linux servers/Databases/MySQL/Tokyo` создает соответствующую подгруппу внутри вложенных родительских групп `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

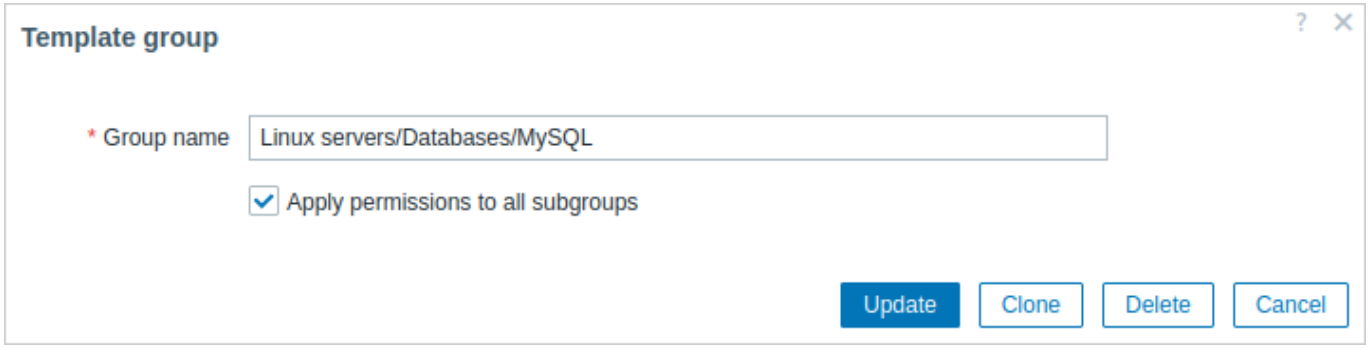
При создании подгруппы не допускается использование ведущих или завершающих косых черт, а также нескольких косых черт подряд. Экранирование '/' не поддерживается.

Перед созданием подгруппы не требуется предварительно создавать родительскую группу или группы шаблонов. Вы можете начать либо с создания подгруппы (например, `Linux servers/Databases`), либо с любой родительской группы шаблонов (в нашем примере — `Linux servers`). Если вы начнете с создания подгруппы, родительская группа или группы шаблонов автоматически **не** будут созданы.

Права доступа к группам шаблонов

- При создании подгруппы в существующей родительской группе шаблонов (например, при создании `Linux servers/Databases`, когда `Linux servers` уже существует), группе пользователей наследуются права доступа к подгруппе от родительской группы.
- При создании родительской группы шаблонов для существующей подгруппы (например, при создании `Linux servers`, когда `Linux servers/Databases` уже существует), права доступа к родительской группе не задаются.

При редактировании любой группы шаблонов можно также задать дополнительную опцию *Применить права доступа ко всем подгруппам*.



Если установить этот флажок и нажать *Обновить*, к всем текущим и будущим подгруппам редактируемой группы шаблонов будет применен тот же уровень прав доступа.

Таким образом, если каким-либо группам пользователей были назначены разные права доступа к подгруппам редактируемой группы шаблонов, установка флажка предоставит всем текущим и будущим подгруппам те же права пользователей, что и у редактируемой группы.

Обратите внимание, что эта опция не сохраняется в базе данных и переопределяет существующие права доступа. Любые изменения, внесенные с помощью этой опции, можно отменить только вручную.

Связывание/отвязка

Обзор

Присоединением называется процесс, при котором шаблоны применяются к узлам сети, тогда как отсоединение — удаление связи шаблона с узлом сети.

Привязка шаблона

Чтобы привязать шаблон к узлу сети, выполните следующие действия:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Нажмите на нужный узел сети.
3. Начните вводить имя шаблона в поле *Шаблоны*. Появится список соответствующих шаблонов; прокрутите его вниз и выберите нужный шаблон. Также можно нажать *Выбрать* рядом с полем и выбрать один или несколько шаблонов из списка во всплывающем окне.
4. Нажмите *Добавить/Обновить* в форме атрибутов узла сети.

Теперь узел сети будет содержать все сущности шаблона. К ним относятся элементы данных, триггеры, графики, правила обнаружения низкого уровня, веб-сценарии, а также панели мониторинга.

Attention:
Привязка нескольких шаблонов к одному узлу сети завершится ошибкой, если эти шаблоны содержат элементы данных с одинаковыми ключами элементов данных. Поскольку триггеры и графики используют элементы данных, их также нельзя привязать к одному узлу сети из нескольких шаблонов, если используются идентичные ключи элементов данных.

При добавлении сущностей (элементов данных, триггеров и т. д.) из шаблона:

- ранее существовавшие на узле сети идентичные сущности обновляются как сущности шаблона, и **все существующие пользовательские настройки сущности на уровне узла сети теряются**;
- сущности из шаблона добавляются;
- все непосредственно привязанные сущности, которые до привязки шаблона существовали только на узле сети, остаются без изменений.

В списках перед всеми сущностями шаблона теперь отображается имя шаблона, указывающее на их принадлежность к определенному шаблону. Само имя шаблона (отображаемое серым цветом) является ссылкой, позволяющей открыть список этих сущностей на уровне шаблона.

Note:
Для некоторых элементов данных, например внешних проверок, проверок с использованием HTTP-агента, простых проверок, проверок SSH и проверок Telnet, интерфейс узла сети является необязательным. Если на момент привязки шаблона для узла сети не определен интерфейс, эти элементы данных будут добавлены без интерфейса. Если позднее добавить интерфейс узла сети, он не будет автоматически назначен уже существующим элементам данных. Чтобы одновременно назначить вновь добавленный интерфейс узла сети всем элементам данных шаблона, отвяжите шаблон от узла сети, а затем привяжите его снова. Чтобы сохранить историю элементов данных, используйте параметр *Отвязать*, а не *Отвязать и очистить*.

Если перед какой-либо сущностью не указано имя шаблона, это означает, что она существовала на узле сети до привязки и не была добавлена шаблоном.

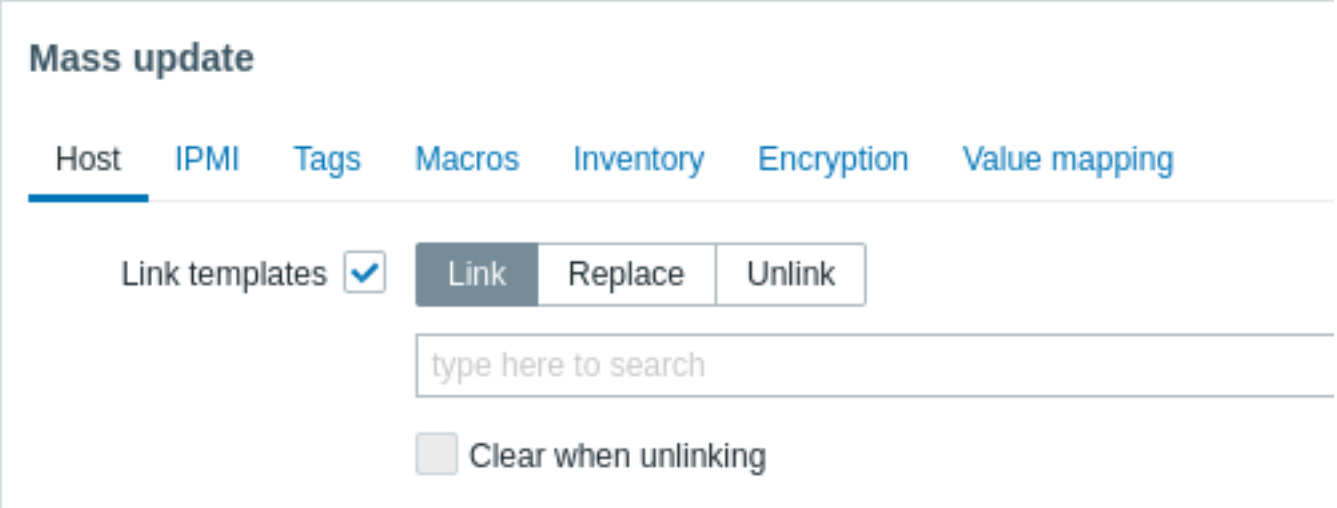
Критерии уникальности объекта

При добавлении объектов (элементов данных, триггеров и т.п.) из шаблона важно знать: какие из этих объектов уже существуют у узла сети и должны быть обновлены, а какие объекты различаются. Критериями уникальности при принятии решения на сходство/различия являются:

- для элементов данных — ключ элемента данных;
- для триггеров — имя и выражение триггера;
- для пользовательских графиков — имя графика и его элементы данных.

Привязка шаблонов к нескольким узлам сети

Чтобы обновить привязку шаблонов для нескольких узлов сети, в разделе *Сбор данных* > *Узлы сети* выберите нужные узлы сети, установив флажки, затем нажмите **Массовое обновление** под списком и выберите *Привязать шаблоны*:



Чтобы привязать дополнительные шаблоны, начните вводить имя шаблона в поле автодополнения, пока не появится раскрывающийся список с подходящими шаблонами. Прокрутите список вниз и выберите шаблон для привязки.

Опция *Заменить* позволяет привязать новый шаблон, одновременно отвязав все шаблоны, которые ранее были привязаны к узлам сети. Опция *Отвязать* позволяет указать, какие шаблоны следует отвязать. Опция *Очистить при отвязывании* позволяет не только отвязать ранее привязанные шаблоны, но и удалить все унаследованные от них сущности (элементы данных, триггеры и т. д.).

Note:
Zabbix предлагает большой набор предопределенных шаблонов. Вы можете использовать их в качестве справочных, но не рекомендуется применять их без изменений в рабочей среде, поскольку они могут содержать слишком много элементов данных и слишком часто опрашивать данные. Если вы хотите их использовать, настройте их в соответствии с вашими реальными потребностями.

Изменение соединённых объектов

Если вы попытаетесь изменить элемент данных или триггер, который присоединён из шаблона, вы можете увидеть, что многие ключевые опции недоступны для редактирования. Такой подход имеет смысл, так как идея шаблонов — чтобы изменять объекты в одно касание на уровне шаблона. Тем не менее, вы всё ещё можете, например, активировать/деактивировать элемент данных на отдельном узле сети, указать интервал обновления, срок хранения истории и некоторые другие параметры.

Если вы хотите изменить полностью объект, вы должны отредактировать его на уровне шаблона (ссылка на уровень шаблона отображается в имени диалога), но имейте ввиду, что все эти изменения будут влиять на все узлы сети, которые соединены с этим шаблоном.

> **Attention:**
> Любые настройки объектов, реализованные на уровне шаблона, перезапишут ранее сделанные настройки этих объектов на уровне узлов сети.

Отвязка шаблона

Чтобы отвязать шаблон от узла сети, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Щелкните нужный узел сети и найдите поле *Шаблоны*.
3. Нажмите *Отвязать* или *Отвязать и очистить* рядом с шаблоном, чтобы отвязать шаблон.
4. Нажмите *Обновить* в форме атрибутов узла сети.

При выборе параметра *Отвязать* будет просто удалена связь с шаблоном, при этом все его сущности останутся у узла сети. Это включает элементы данных, триггеры, графики, правила обнаружения низкого уровня и веб-сценарии, но не включает панели мониторинга. Обратите внимание, что карты значений и теги, унаследованные от связанных шаблонов, также будут удалены.

При выборе параметра *Отвязать и очистить* будут удалены и связь с шаблоном, и все его сущности (элементы данных, триггеры и т. д.).

Вложенность

Обзор

Наследование — это способ одного шаблона включать в себя один или несколько других шаблонов.

Иногда имеет смысл разделить объекты отдельных шаблонов на различные сервисы, приложения и прочее, в итоге у вас может получиться несколько шаблонов, которые нужно присоединить к довольно большому количеству узлов сети. Для простоты картины, возможно вложить в один шаблон несколько других шаблонов.

Преимущества наследования в том, что вы должны будете присоединить к узлу сети только один шаблон («основной», родительский шаблон), и этот узел сети автоматически унаследует все объекты из присоединённых шаблонов («вложенные», дочерние шаблоны). Например, если мы присоединим шаблоны *T1* и *T2* к шаблону *T3*, мы дополняем *T3* объектами из *T1* и *T2*, но не наоборот. Если мы присоединим шаблон *T1* к шаблонам *T2* и *T3*, мы дополняем шаблоны *T2* и *T3* объектами из *T1*.

Настройка вложенных шаблонов

Чтобы связать шаблоны, необходимо взять существующий шаблон (или создать новый), а затем:

1. Откройте форму настройки шаблона.
2. Найдите поле *Templates*.
3. Нажмите *Select*, чтобы открыть всплывающее окно *Templates*.
4. Во всплывающем окне выберите нужные шаблоны, а затем нажмите *Select*, чтобы добавить шаблоны в список.
5. Нажмите *Add* или *Update* в форме настройки шаблона.

Таким образом, все сущности настроенного шаблона, а также все сущности связанных шаблонов, теперь будут отображаться в настройке шаблона. Это включает элементы данных, триггеры, графики, правила обнаружения низкого уровня и веб-сценарии, но не включает панели мониторинга. Однако панели мониторинга связанных шаблонов, тем не менее, будут унаследованы узлами сети.

Чтобы отвязать любой из связанных шаблонов, нажмите *Unlink* или *Unlink and clear* в форме настройки шаблона, а затем нажмите *Update*.

Параметр *Unlink* просто удалит связь с связанным шаблоном, не удаляя при этом ни одну из его сущностей (элементы данных, триггеры и т. д.).

Параметр *Unlink and clear* удалит как связь с связанным шаблоном, так и все его сущности (элементы данных, триггеры и т. д.).

Массовое обновление

Обзор

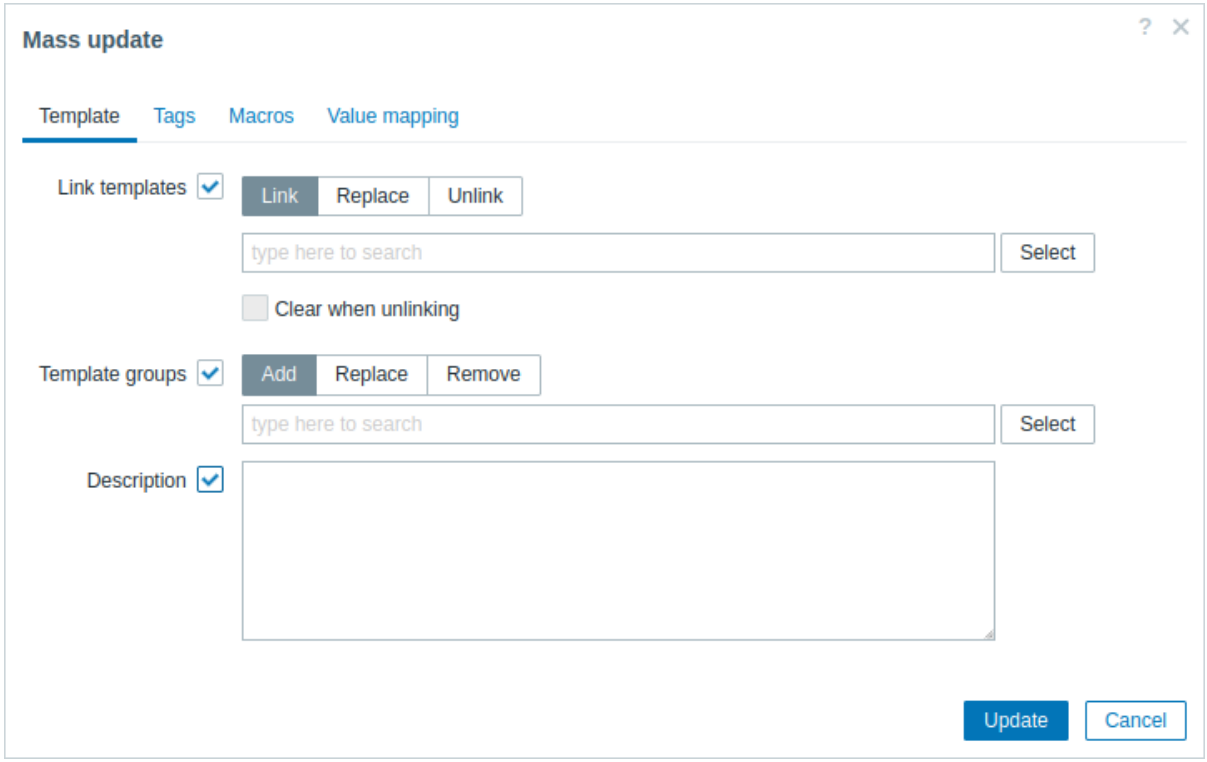
Иногда вы можете захотеть изменить какой-либо атрибут для нескольких шаблонов одновременно. Вместо того, чтобы открывать каждый отдельный шаблон для редактирования, вы можете воспользоваться функцией массового обновления.

Массовое обновление

Чтобы массово обновить некоторые шаблоны, выполните следующие действия:

1. Отметьте флажки перед шаблонами, которые вы хотите обновить, в списке шаблонов.
2. Нажмите *Массовое обновление* под списком.
3. Перейдите на вкладку с нужными атрибутами (*Шаблон*, *Теги*, *Макросы* или *Преобразование значений*).
4. Отметьте флажки нужных атрибутов для обновления и введите для них новое значение.

Вкладка **Шаблон** содержит общие параметры массового обновления шаблонов.



При выборе соответствующей кнопки для обновления *Связать шаблоны* доступны следующие параметры:

- *Связать* — укажите дополнительные шаблоны для связывания;
- *Заменить* — укажите шаблоны для связывания, одновременно отменив связь со всеми ранее связанными шаблонами;
- *Отменить связь* — укажите шаблоны, связь с которыми необходимо отменить.

Чтобы указать шаблоны для связывания или отмены связи, начните вводить имя шаблона в поле автозаполнения, пока не появится раскрывающийся список с подходящими шаблонами. Прокрутите список вниз, чтобы выбрать нужные шаблоны.

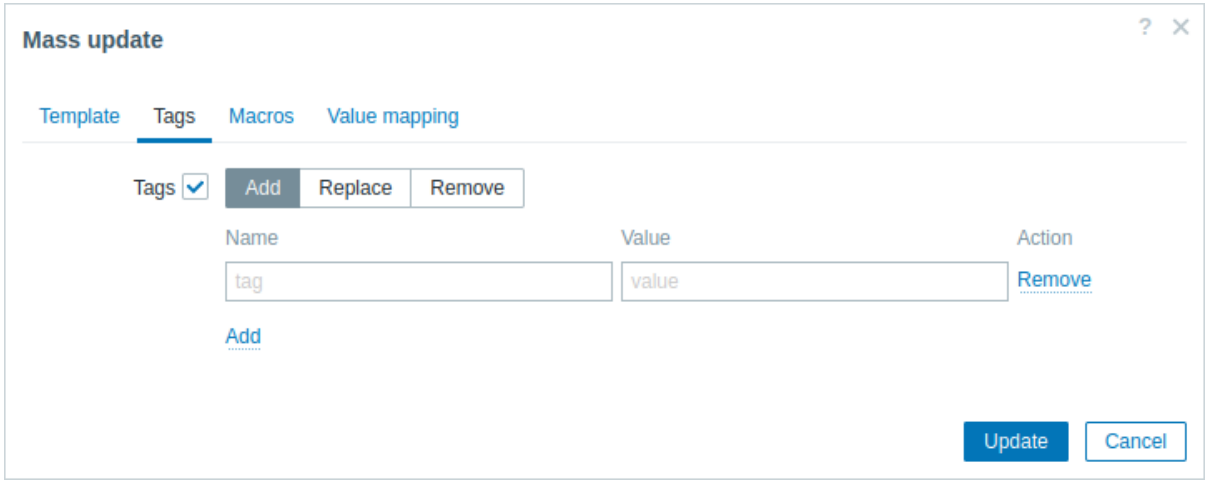
Параметр *Очистить при отмене связи* позволяет не только отменить связь с ранее связанными шаблонами, но и удалить все унаследованные от них сущности (элементы данных, триггеры, графики и т. д.).

При выборе соответствующей кнопки для обновления *Группы шаблонов* доступны следующие параметры:

- *Добавить* — укажите дополнительные группы шаблонов из существующих или введите для шаблонов совершенно новые группы шаблонов;
- *Заменить* — удалите шаблон из всех существующих групп шаблонов и замените их группой или группами, указанными в этом поле (существующими или новыми группами шаблонов);
- *Удалить* — удалите определенные группы шаблонов из шаблонов.

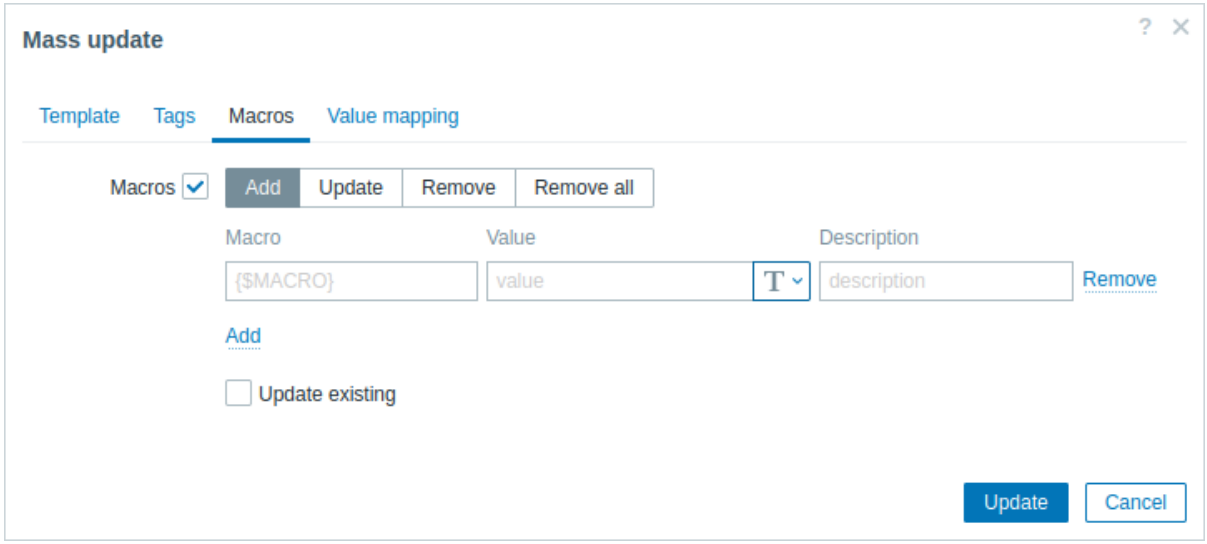
Эти поля поддерживают автозаполнение: начните вводить текст, и появится раскрывающийся список с подходящими группами шаблонов. Если группа шаблонов новая, она также появится в раскрывающемся списке и будет обозначена строкой *(новая)*. Прокрутите список вниз, чтобы выбрать нужную группу.

Вкладка **Теги (Tags)** позволяет вам массово обновлять теги на уровне шаблона.



В тегах поддерживаются пользовательские макросы, макросы {INVENTORY.*}, а также {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}. Обратите внимание, теги с одинаковыми именами, но с разными значениями, не считаются «дубликатами», и их можно добавлять на один шаблон.

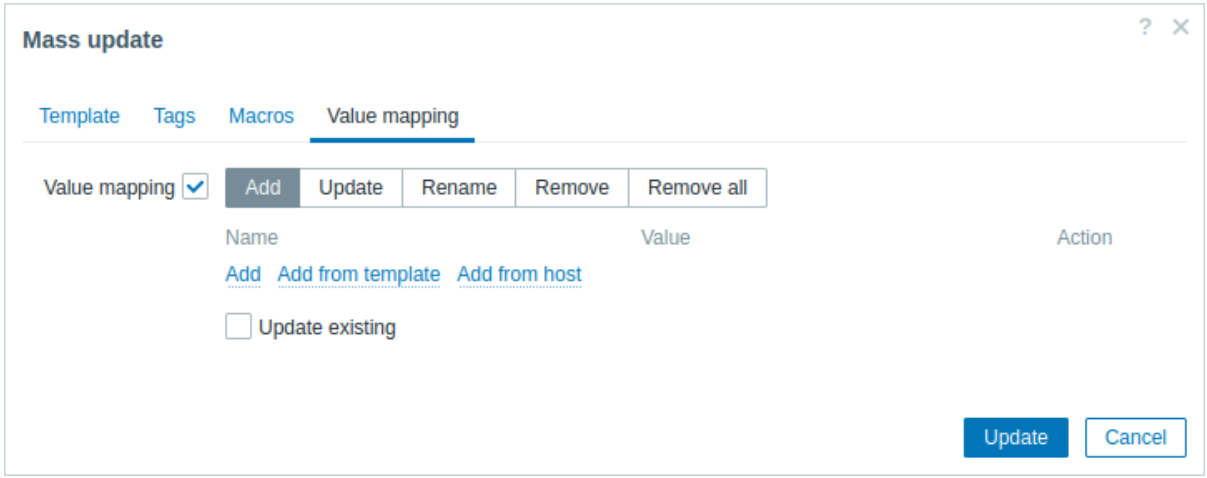
Вкладка **Макросы** позволяет массово обновлять макросы на уровне шаблона.



При выборе соответствующей кнопки доступны следующие варианты обновления макросов:

- *Добавить* — указать дополнительные пользовательские макросы для шаблонов. Если установлен флажок *Обновить существующие*, для указанного имени макроса будут обновлены значение, тип и описание. Если флажок не установлен и макрос с таким именем уже существует в шаблоне или шаблонах, он не будет обновлен.
- *Обновить* — заменить значения, типы и описания макросов, указанных в этом списке. Если установлен флажок *Добавить отсутствующие*, макрос, которого ранее не было в шаблоне, будет добавлен как новый макрос. Если флажок не установлен, будут обновлены только макросы, которые уже существуют в шаблоне.
- *Удалить* — удалить указанные макросы из шаблонов. Если установлен флажок *Кроме выбранных*, будут удалены все макросы, кроме указанных в списке. Если флажок не установлен, будут удалены только макросы, указанные в списке.
- *Удалить все* — удалить все пользовательские макросы из шаблонов. Если флажок *Я подтверждаю удаление всех макросов* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления всех макросов.

Вкладка **Сопоставление значений** позволяет массово обновлять сопоставления значений.



Для обновления сопоставлений значений доступны кнопки со следующими параметрами:

- *Добавить* - добавить сопоставления значений в шаблоны. Если выбрать *Обновить существующие*, будут обновлены все свойства сопоставления значений с этим именем. В противном случае, если сопоставление значений с таким именем уже существует, оно не будет обновлено.
- *Обновить* - обновить существующие сопоставления значений. Если выбрать *Добавить отсутствующие*, сопоставление значений, которого ранее не было в шаблоне, будет добавлено как новое сопоставление значений. В противном случае будут обновлены только сопоставления значений, которые уже существуют в шаблоне.
- *Переименовать* - присвоить новое имя существующему сопоставлению значений.
- *Удалить* - удалить указанные сопоставления значений из шаблонов. Если выбрать *Кроме выбранных*, будут удалены все сопоставления значений, **кроме** указанных.
- *Удалить все* - удалить все сопоставления значений из шаблонов. Если флажок *Я подтверждаю удаление всех сопоставлений значений* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления.

Параметры *Добавить из шаблона* и *Добавить из узла сети* доступны для операций добавления и обновления сопоставлений значений. Они позволяют выбрать сопоставления значений соответственно из шаблона или узла сети.

Когда все необходимые изменения будут внесены, нажмите *Update*. Атрибуты будут соответствующим образом обновлены для всех выбранных шаблонов.

Узлы сети и группы узлов сети

Что такое ”узел сети”?

В Zabbix ”узел сети” — это любое физическое или виртуальное устройство, приложение, служба или любая другая логически связанная совокупность контролируемых параметров.

Создание узлов сети — одна из первых задач мониторинга в Zabbix. Например, если вы хотите отслеживать некоторые параметры на сервере ”x”, сначала необходимо создать узел сети, например ”Server X”, а затем можно добавить к нему элементы данных для мониторинга.

Узлы сети организуются в группы узлов сети.

Перейдите к:

- Мастер узлов сети для пошагового интерфейса, который поможет создать и настроить новый или существующий узел сети.
- Настройка узла сети для классического способа создания и настройки узла сети.

Мастер узла сети

Обзор

Мастер узлов сети — это пошаговый интерфейс с подсказками для настройки вашего объекта мониторинга (устройства, приложения, службы и т. д.) в Zabbix. Он проведет вас через следующие шаги:

- Выбор шаблона
- Создание или выбор узла сети
- Установка агента Zabbix или agent 2
- Добавление интерфейса узла сети

- Применение дополнительной конфигурации к вашему объекту мониторинга или узлу сети Zabbix (если это требуется шаблоном)

Чтобы открыть Мастер узлов сети в веб-интерфейсе Zabbix, выполните следующие действия:

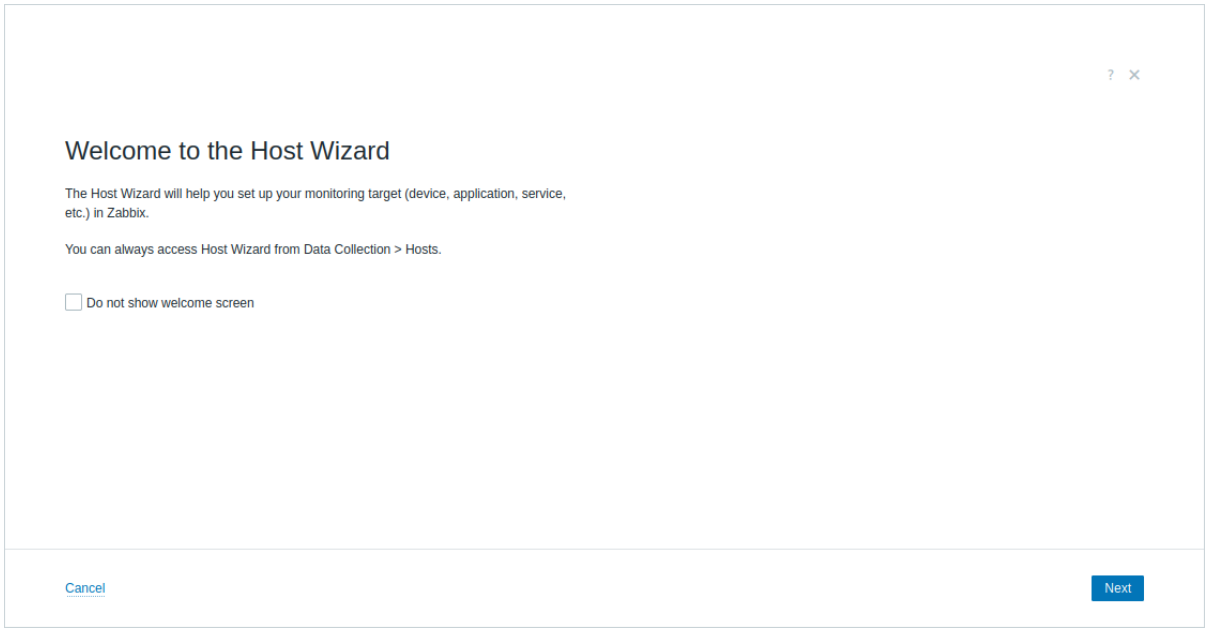
1. Перейдите в: *Сбор данных* > *Узлы сети*.
2. Нажмите *Мастер узлов сети* в правом верхнем углу экрана.
3. Следуйте инструкциям мастера.

Вы также можете изменить существующий узел сети, щелкнув по его имени (или по значку с тремя точками рядом с ним) в различных разделах веб-интерфейса, чтобы открыть Мастер узлов сети из меню узла сети. Обратите внимание, что при редактировании существующего узла сети некоторые шаги пропускаются, если он уже включает конфигурацию, требуемую выбранным шаблоном.

В качестве альтернативы вы можете использовать классический способ создания и настройки узла сети.

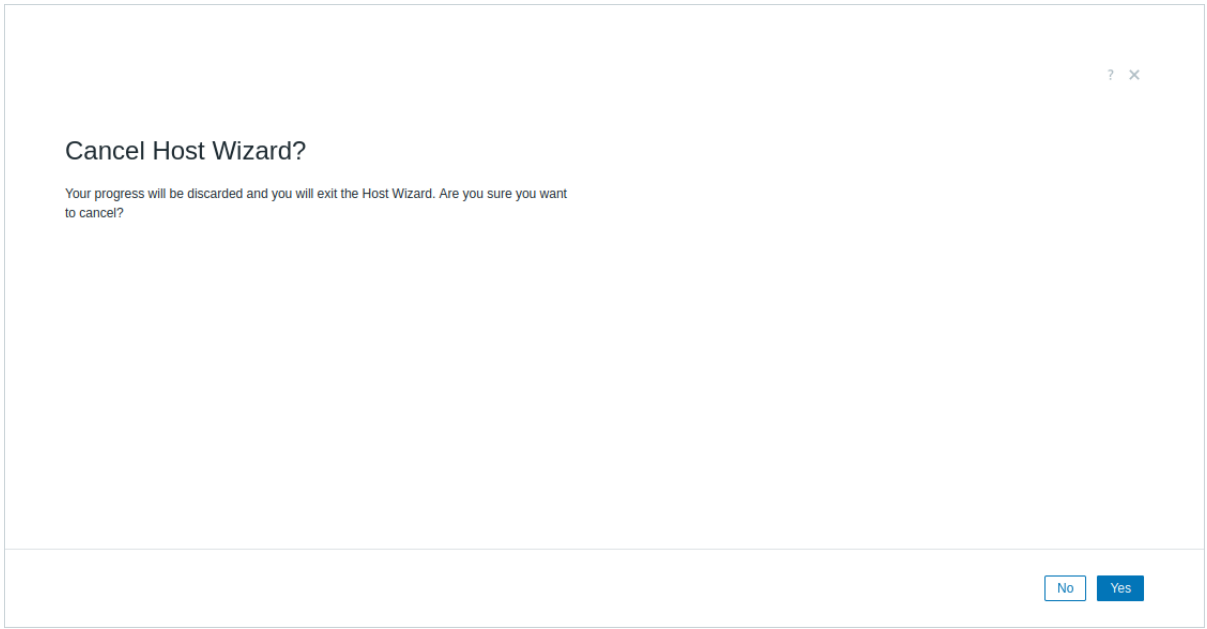
Добро пожаловать в мастер узла сети

Когда вы запускаете мастер узла сети, отображается экран приветствия:



Чтобы в будущем пропускать этот экран, установите флажок *Не показывать экран приветствия* и нажмите *Далее*.

Если вы начнете настраивать параметры на следующих шагах мастера узла сети и попытаетесь закрыть мастер, нажав значок закрытия или Esc, появится экран подтверждения:



При нажатии *Да* мастер узла сети будет закрыт без сохранения изменений. При нажатии *Нет*, Esc или значка закрытия будет выполнен возврат к последнему шагу.

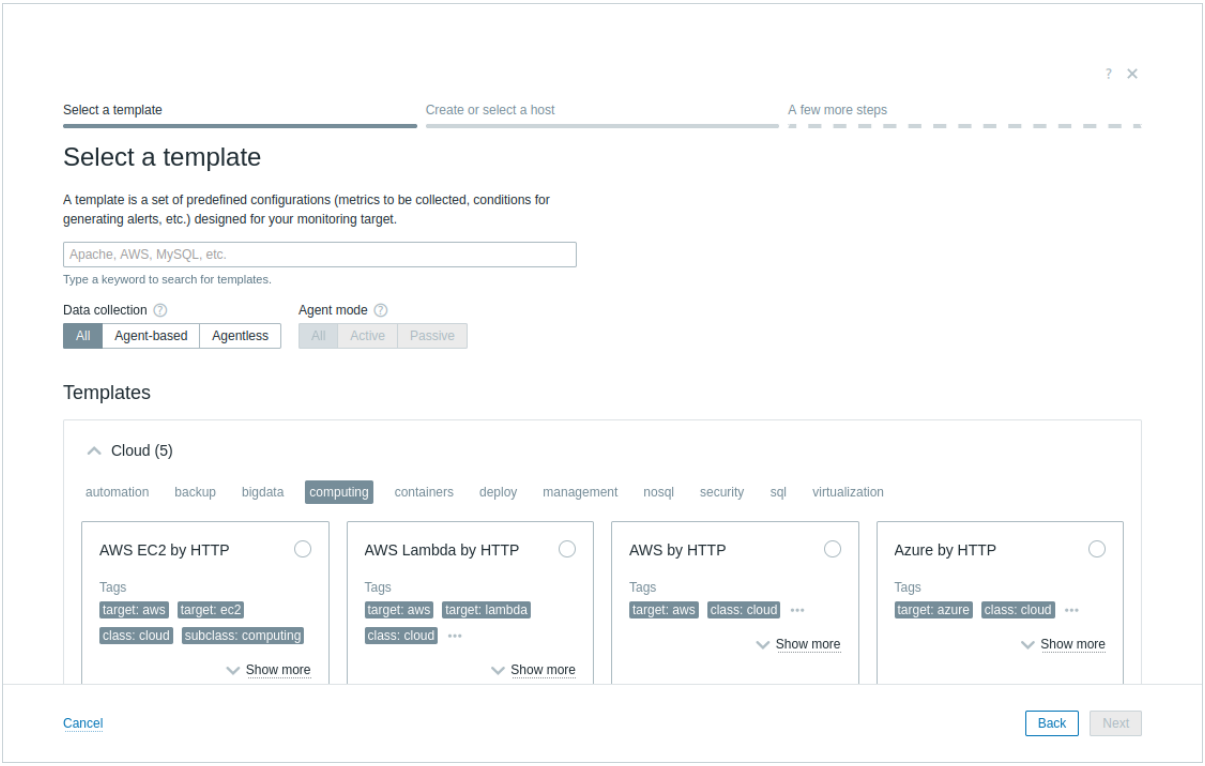
Выбор шаблона

Первым шагом настройки мониторинга целевого объекта является выбор шаблона — набора предварительно заданных конфигураций (собираемые метрики, условия формирования оповещений и т. д.), предназначенных для мониторинга вашего объекта.

Готовые шаблоны Zabbix предлагают различные предварительно заданные конфигурации мониторинга операционных систем, приложений, баз данных, сетевых устройств, сервисов и других объектов.

На этом шаге можно:

- Просматривать шаблоны по классу (облако, база данных, сеть и т. д.) и подклассу (например, автоматизация, резервное копирование), которые основаны на тегах шаблонов.
- Искать шаблоны по ключевым словам в названии шаблона или названиях и значениях тегов шаблона.
- Фильтровать шаблоны по способу сбора данных (на основе агента или без агента; шаблоны на основе агента содержат как минимум один элемент данных агента Zabbix).
- Фильтровать шаблоны по режиму агента (активный или пассивный; активные шаблоны содержат как минимум один элемент данных агента Zabbix (активный), а пассивные шаблоны содержат как минимум один пассивный элемент данных; подробную информацию см. в разделе Пассивные и активные проверки агента).
- При использовании мастера узла сети для настройки существующего узла сети также фильтровать шаблоны, уже связанные с узлом сети.



Attention:
При выборе шаблона может отображаться следующее сообщение: *Некоторые шаблоны (n) несовместимы с мастером узла сети. Узнайте, как обновить их. Пользовательские шаблоны не поддерживаются.* Это означает, что некоторые готовые шаблоны еще несовместимы с мастером узла сети и требуют обновления; инструкции см. в разделе обновление шаблонов.

Выбранный шаблон определяет следующие шаги в мастере узла сети. Например, если выбрать шаблон *MySQL by Zabbix agent*, использующий агент Zabbix для сбора данных, мастер проведет вас через процесс установки агента.

Создание или выбор узла сети

Шаблон необходимо связать с узлом сети — сущностью в Zabbix, представляющей объект мониторинга. После связывания узел сети получает все сущности шаблона, такие как элементы данных (собираемые метрики) и триггеры (условия для формирования оповещений), необходимые для мониторинга.

Каждый узел сети также должен принадлежать как минимум к одной группе узлов сети, которая используется для организации узлов сети и назначения пользовательских прав доступа к ним.

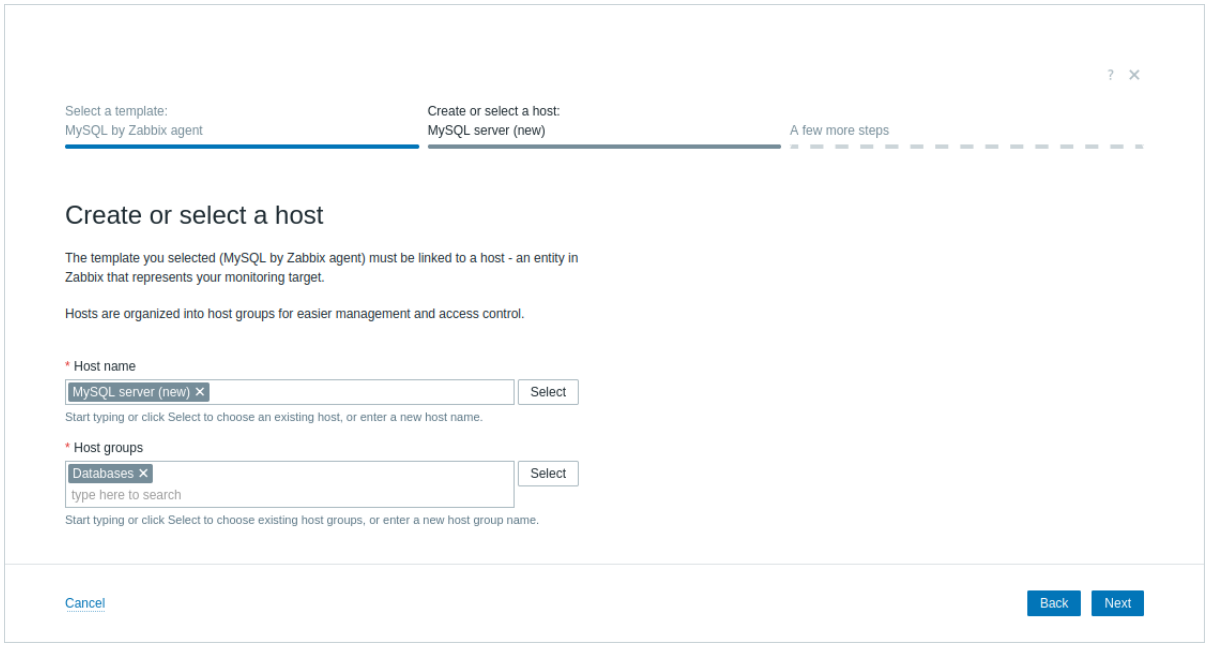
На этом шаге можно:

- Создать новый узел сети и группу узлов сети, указав имена для обоих объектов.

- Создать новый узел сети, указав его имя, и выбрать существующую группу узлов сети, к которой будет добавлен новый узел сети.
- Выбрать существующий узел сети; также можно добавить этот узел сети в дополнительные группы узлов сети, не удаляя его из текущих групп.

Note:
Выбор существующего узла сети не удаляет и не перезаписывает его текущую конфигурацию, если это явно не указано на последующих шагах. Обнаруженные узлы сети выбрать нельзя.

Например, можно создать новый узел сети с именем *MySQL server*, представляющий локально установленный сервер MySQL, который требуется мониторить. С этим узлом сети будет связан ранее выбранный шаблон *MySQL by Zabbix agent*. Кроме того, можно выбрать существующую группу узлов сети *Databases* (или создать новую, например *MySQL servers*), чтобы объединить этот узел сети с другими узлами сети, связанными с базами данных, и впоследствии упростить управление правами доступа.



Установите агент Zabbix

Шаблоны на основе агента (например, *MySQL by Zabbix agent*) требуют установки агента Zabbix или агента 2 - процесса, развернутого на контролируемом узле сети для активного мониторинга локальных ресурсов и приложений (если это требуется шаблоном).

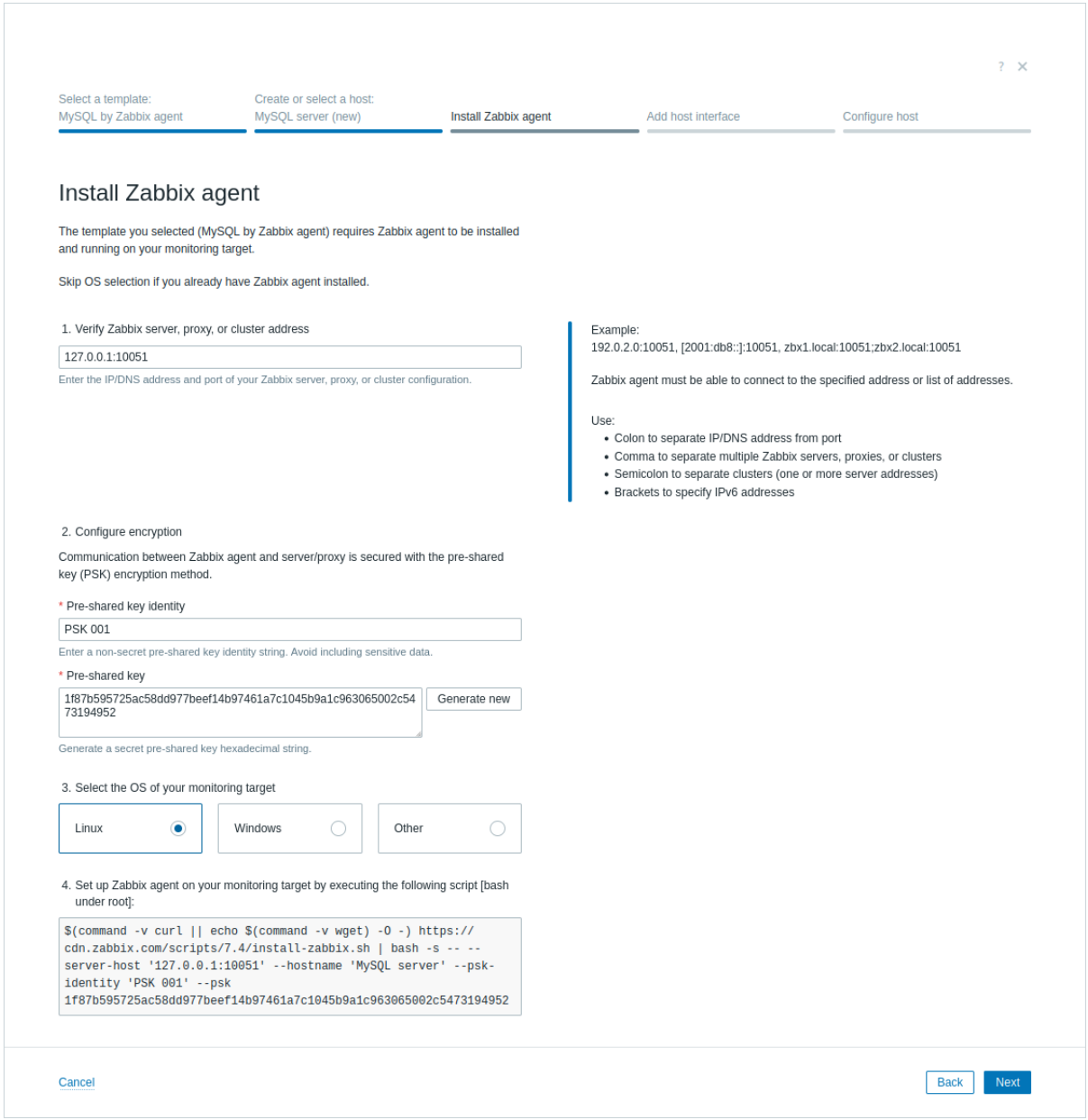
На этом шаге:

- Проверьте адрес сервера Zabbix, прокси или кластера (например, 192.0.2.1:10051). Это значение будет использоваться для настройки параметров Server и ServerActive агента Zabbix.
- Введите новый уникальный и несекретный идентификатор предварительно установленных ключей (например, *PSK 001* или *mysql-agent-psk1*) и сгенерируйте предварительно установленный ключ. Эти значения будут использоваться для настройки шифрования с помощью предварительно установленных ключей (PSK) для узла сети Zabbix (например, *сервера MySQL*) и агента Zabbix.

Attention:
При редактировании существующего узла сети конфигурация PSK перезапишет все существующие настройки шифрования узла сети.

- Выберите операционную систему компьютера, на котором размещен контролируемый узел сети.
- Установите агент Zabbix в этой системе, выполнив на ней предоставленный скрипт или следуя инструкциям по установке.

Например, можно задать для параметра *Pre-shared key identity* значение *PSK 001* и сгенерировать новый *Pre-shared key*. Затем можно выбрать *Linux* в качестве операционной системы и выполнить предоставленный скрипт в этой системе. После установки агента Zabbix вернитесь к мастеру узла сети.

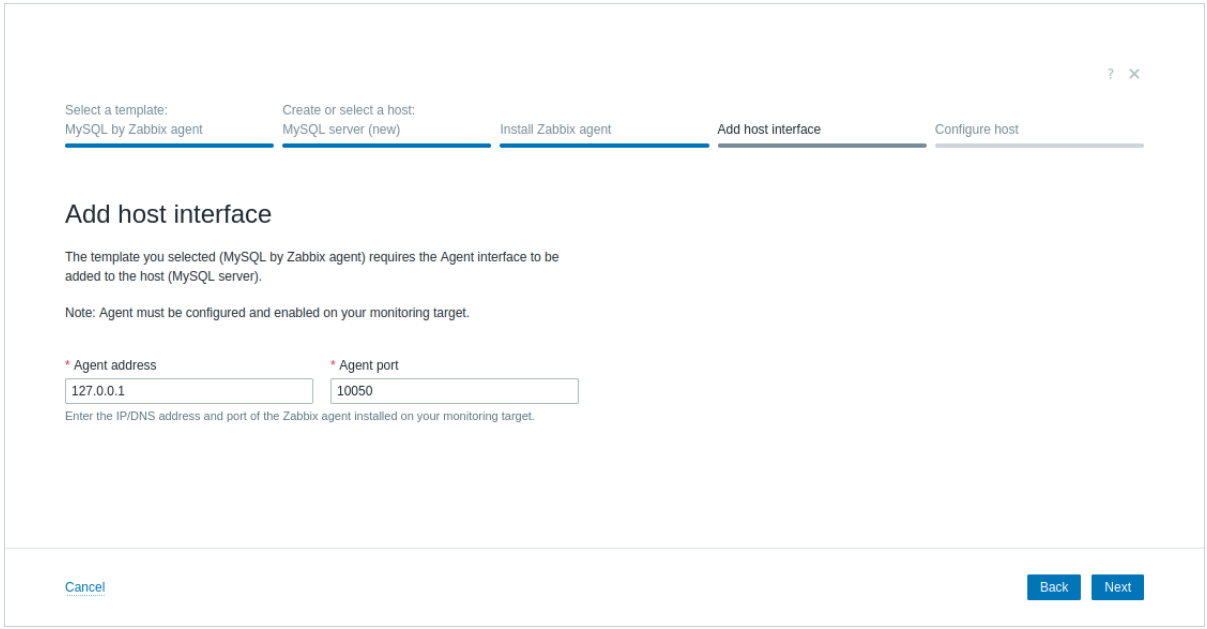


Добавление интерфейса узла сети

Интерфейс узла сети определяет, как сервер Zabbix подключается к целевому объекту мониторинга по сети. Он задает параметры подключения, такие как IP-адрес, DNS-имя, порт и тип интерфейса (агент, SNMP, JMX или IPMI), в зависимости от метода сбора данных, требуемого выбранным шаблоном.

На этом шаге введите сведения об интерфейсе узла сети, требуемые выбранным шаблоном.

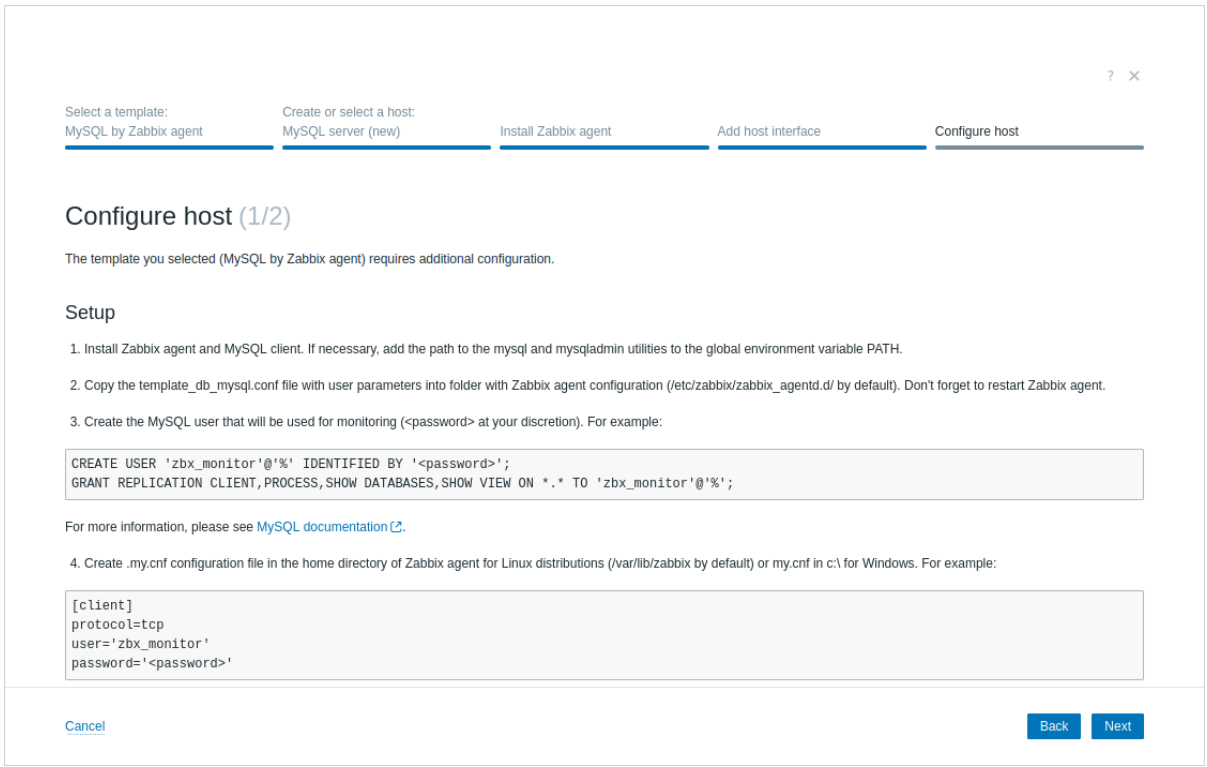
Например, можно использовать адрес агента по умолчанию (127.0.0.1) и порт (10050), если сервер Zabbix, агент Zabbix и сервер MySQL работают локально на одном и том же компьютере.



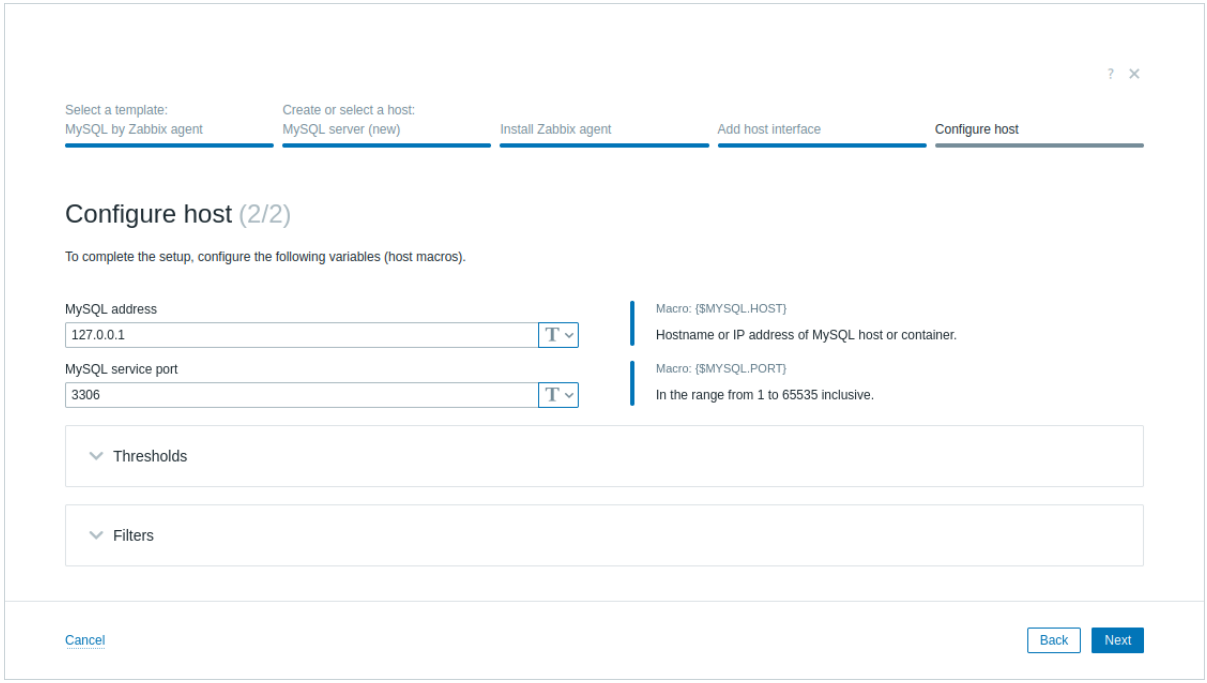
Настройка узла сети

Некоторые шаблоны требуют дополнительной настройки перед созданием узла сети. Она может включать:

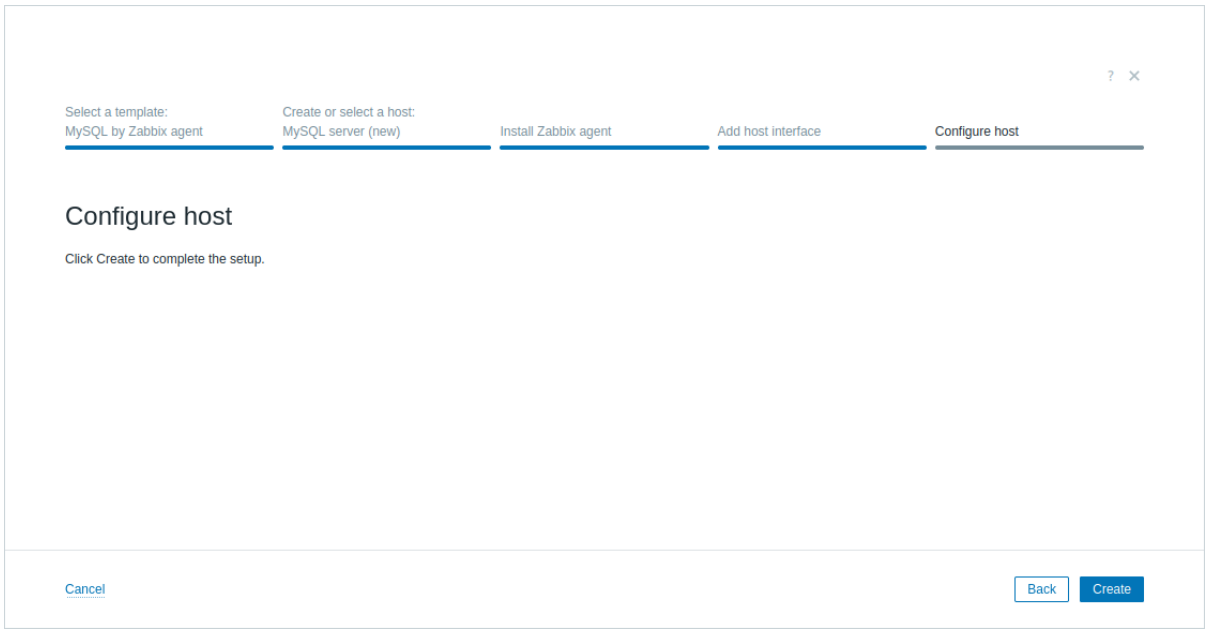
- Ручную настройку объекта мониторинга (например, включение определённых служб, создание пользователей служб или предоставление разрешений):



- Определение макросов узла сети - переменных, которые управляют поведением элементов данных, параметрами подключения, учётными данными для аутентификации и другими параметрами:



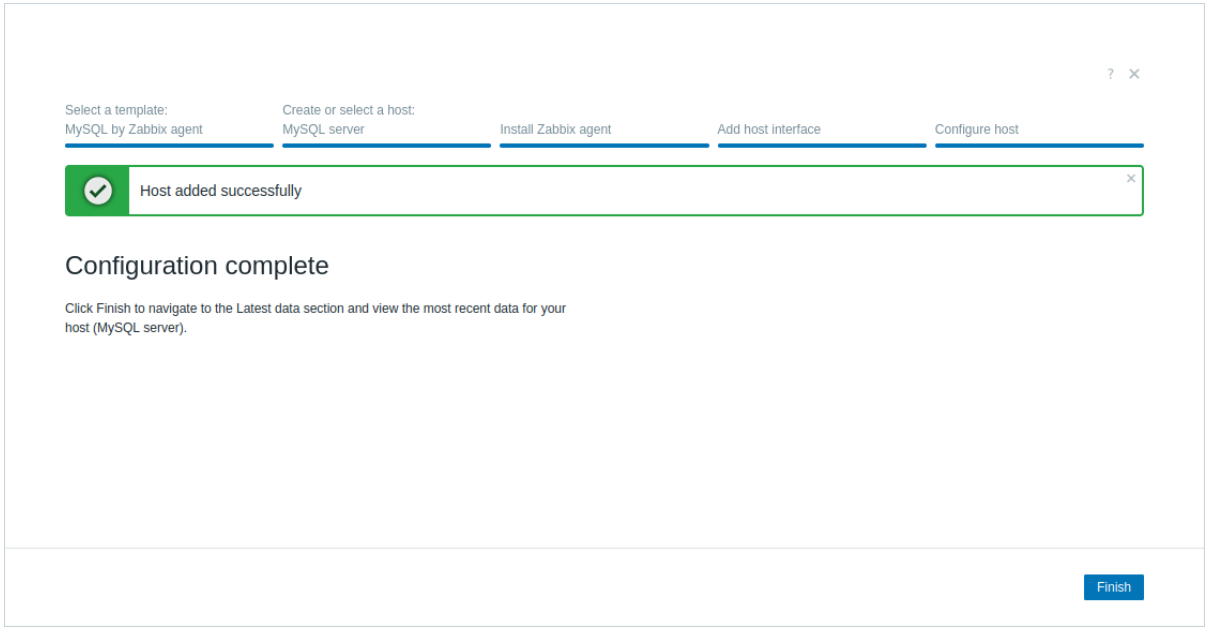
После выполнения всех необходимых шагов настройки нажмите *Создать*, чтобы добавить узел сети в Zabbix (или *Обновить*, чтобы обновить конфигурацию существующего узла сети):



Настройка завершена

На этом этапе Zabbix уже выполняет мониторинг вашего целевого узла сети.

Для новых узлов сети нажмите *Завершить*, чтобы перейти в раздел Последние данные и просмотреть самые свежие данные для вашего узла сети. Для существующих узлов сети нажмите *Завершить*, чтобы закрыть мастер узла сети и вернуться к экрану, с которого он был открыт.



Настройка узла сети

Обзор

Чтобы настроить узел сети в веб-интерфейсе Zabbix, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Узлы сети* или *Мониторинг* > *Узлы сети*.
2. Нажмите *Создать узел сети* в правом верхнем углу экрана (или на имя узла сети, чтобы изменить существующий узел сети).
3. Введите параметры узла сети в форму.

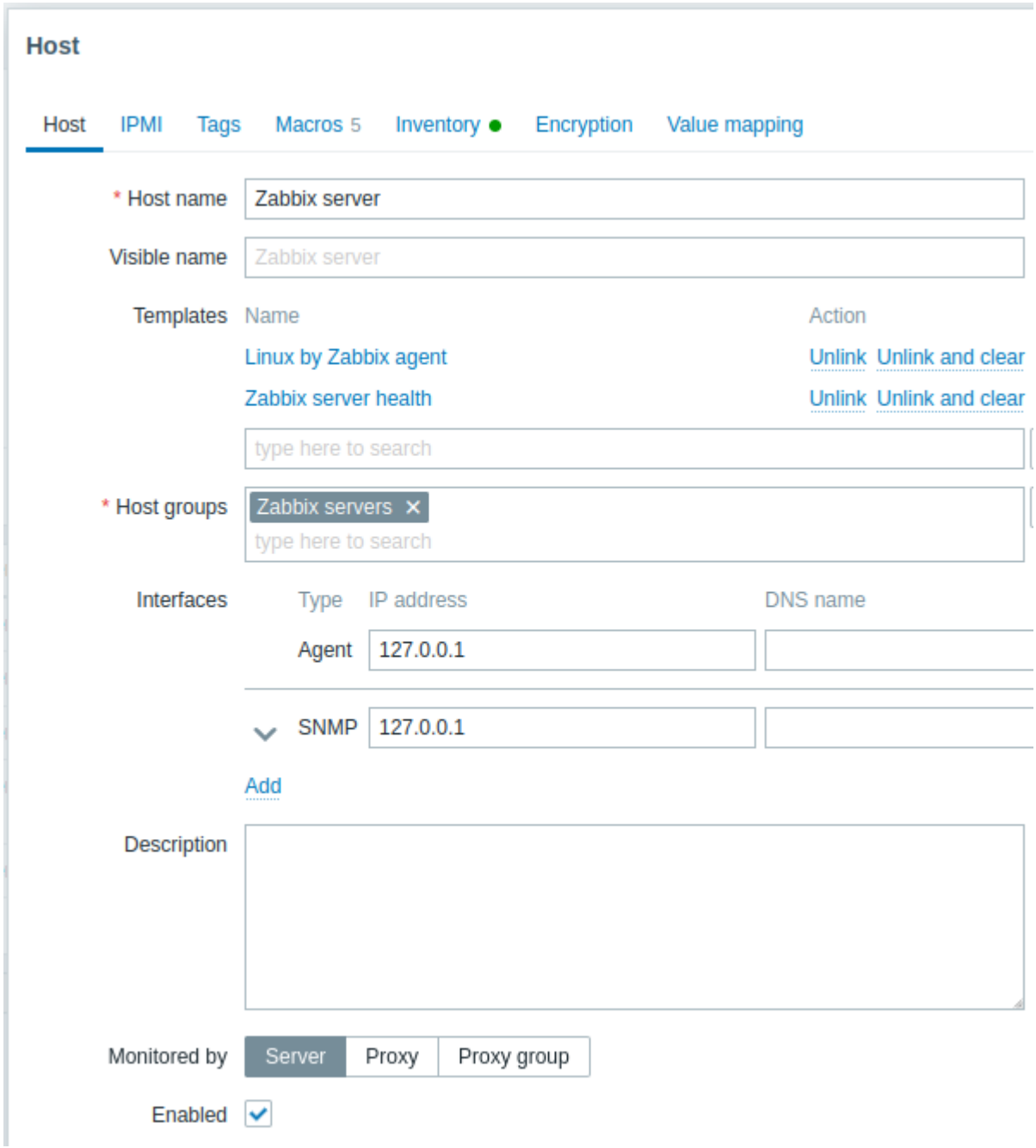
Вы также можете использовать кнопку *Клонировать* в форме настройки существующего узла сети, чтобы создать новый узел сети. Этот узел сети будет иметь все свойства существующего узла сети, включая связанные шаблоны, сущности (элементы данных, триггеры и т. д.) из этих шаблонов, а также сущности, напрямую привязанные к существующему узлу сети.

Обратите внимание, что при клонировании узла сети все сущности шаблона сохранятся в том виде, в котором они изначально определены в шаблоне. Любые изменения этих сущностей, внесенные на уровне существующего узла сети (например, измененный интервал элемента данных, измененное регулярное выражение или добавленные прототипы к правилу низкоуровневого обнаружения), не будут скопированы в новый узел сети; вместо этого они будут такими же, как в шаблоне.

Кроме того, вы можете использовать Мастер узлов сети для настройки узла сети через пошаговый интерфейс с подсказками.

Конфигурация

Вкладка **Узел сети** содержит общие атрибуты узла сети:



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя узла сети	Введите уникальное имя узла сети. Разрешены буквы и цифры, пробелы, точки, дефисы и символы подчеркивания. Однако начальные и конечные пробелы не допускаются. *Примечание:* если на настраиваемом узле сети запущен агент Zabbix, параметр *Hostname* в файле конфигурации агента должен иметь то же значение, что и введенное здесь имя узла сети. Имя, указанное в параметре, необходимо для обработки активных проверок.
Отображаемое имя	Введите уникальное отображаемое имя узла сети. Если указать это имя, оно будет отображаться в списках, на картах и т. д. вместо технического имени узла сети. Этот атрибут поддерживает UTF-8.

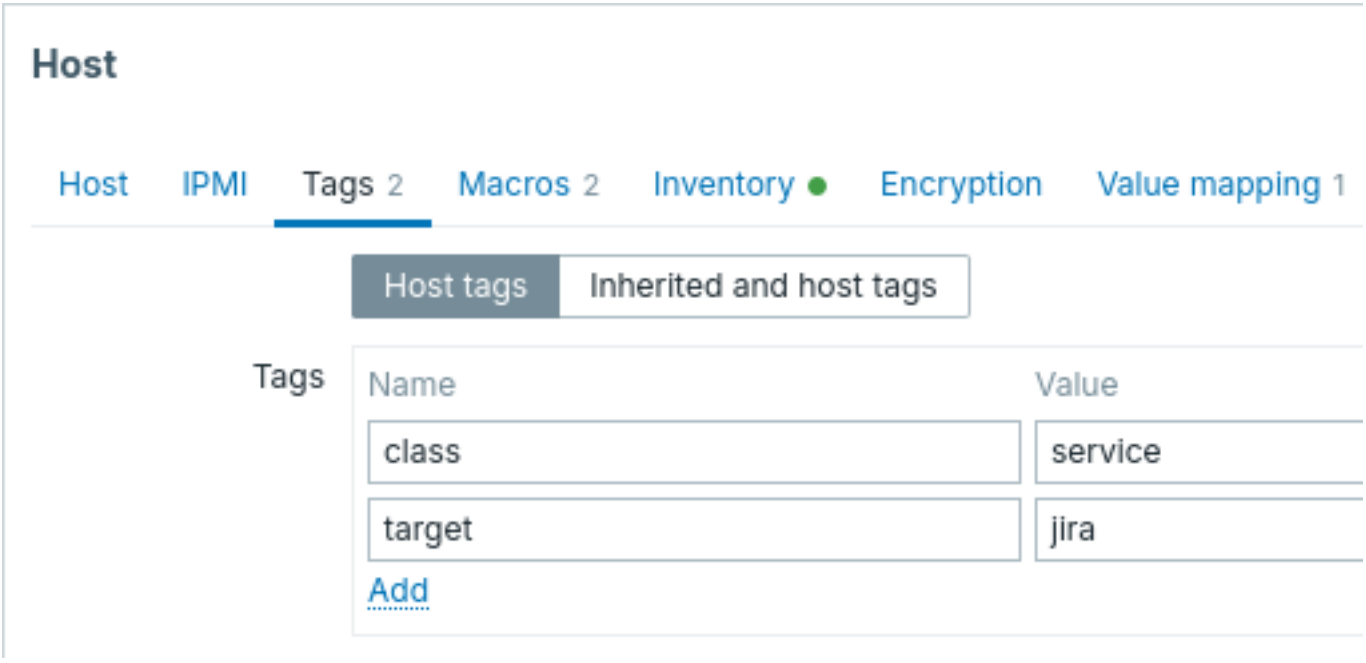
Параметр	Описание
Шаблоны	Свяжите шаблоны с узлом сети. Все сущности (элементы данных, триггеры и т. д.) будут унаследованы от шаблона. Чтобы связать новый шаблон, начните вводить его имя в поле ввода. Появится список соответствующих шаблонов; прокрутите его вниз, чтобы выбрать нужный шаблон. Также можно нажать *Выбрать* рядом с полем и выбрать шаблоны из списка во всплывающем окне. Все выбранные шаблоны будут связаны с узлом сети после сохранения или обновления формы конфигурации узла сети. Чтобы отменить связь с шаблоном, используйте один из двух вариантов в блоке *Связанные шаблоны*: *Отменить связь* — отменить связь с шаблоном, но сохранить его сущности (элементы данных, триггеры и т. д.); *Отменить связь и очистить* — отменить связь с шаблоном и удалить все его сущности (элементы данных, триггеры и т. д.). Имена перечисленных шаблонов являются ссылками, ведущими к форме конфигурации шаблона.
Группы узлов сети	Выберите группы узлов сети, к которым относится узел сети. Узел сети должен относиться как минимум к одной группе узлов сети. Новую группу можно создать и связать с узлом сети, добавив имя несуществующей группы.
Интерфейсы	Для узла сети поддерживаются следующие типы интерфейсов: *Агент*, *SNMP*, *JMX* и *IPMI*. По умолчанию интерфейсы не определены. Чтобы добавить новый интерфейс, нажмите *Добавить* в блоке *Интерфейсы*, выберите тип интерфейса и введите сведения *IP/DNS*, *Подключаться к* и *Порт*. *Примечание:* интерфейсы, используемые в каких-либо элементах данных, нельзя удалить, поэтому ссылка *Удалить* для них недоступна. Значение «IP» или «DNS» интерфейса SNMP также используется для SNMP-трапов. При сопоставлении используется только выбранное значение «IP» или «DNS» в интерфейсе узла сети. Дополнительные сведения о настройке интерфейса SNMP (v1, v2 и v3) см. в разделе Настройка мониторинга SNMP.
IP-адрес	IP-адрес узла сети (необязательно).
Имя DNS	Имя DNS узла сети (необязательно).
Подключаться к	Нажатие соответствующей кнопки указывает серверу Zabbix, что использовать для получения данных от агентов: **IP** — подключаться к IP-адресу узла сети (рекомендуется) **DNS** — подключаться к имени DNS узла сети
Порт	Номер порта TCP/UDP. Значения по умолчанию: 10050 для агента Zabbix, 161 для агента SNMP, 12345 для JMX и 623 для IPMI.
По умолчанию	Установите переключатель, чтобы задать интерфейс по умолчанию.
Описание	Введите описание узла сети.
Мониторинг выполняется	Выберите, кем выполняется мониторинг узла сети: **Сервер** — мониторинг узла сети выполняется сервером Zabbix; **Прокси** — мониторинг узла сети выполняется автономным прокси; **Группа прокси** — мониторинг узла сети выполняется группой прокси.
Прокси	Отображается имя назначенного прокси (только если сервер Zabbix назначил прокси из выбранной группы прокси). Это поле отображается только если мониторинг узла сети выполняется группой прокси.

Параметр	Описание
Включен	Если флажок установлен, узел сети включен и готов к мониторингу. Если флажок снят, узел сети отключен и не мониторится: Для пассивных запросов данных, инициированных сервером Zabbix/прокси (например, агент Zabbix, агент SNMP, простые проверки), мониторинг отключается после синхронизации конфигурации. Триггеры и действия, связанные с узлом сети, также отключаются только после перезагрузки конфигурации. Для активных проверок агента Zabbix мониторинг прекращается в течение периода (примерно 5 секунд), за который агент Zabbix получает информацию об отключении узла сети. В течение этого короткого интервала узел сети продолжит локально собирать данные для активных проверок и пытаться отправлять их на сервер/прокси; однако, поскольку узел сети помечен как *Отключен*, сервер/прокси отклонит эти данные. При отключении узла сети его элементы данных немедленно удаляются из кэша истории (за исключением их последних значений, которые сохраняются для журналов).

Вкладка **IPMI** содержит атрибуты управления IPMI.

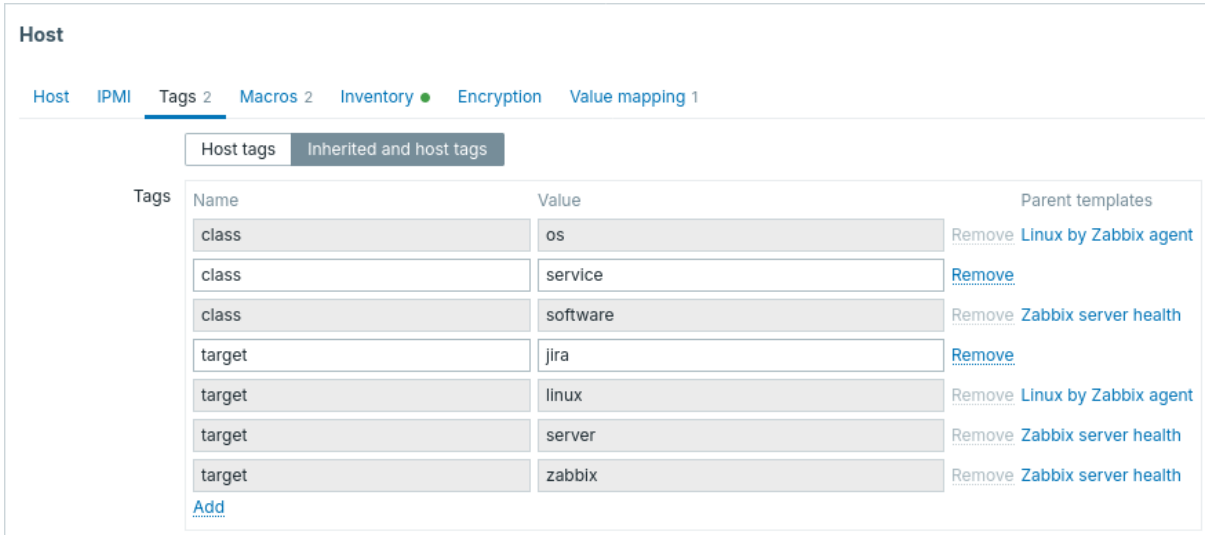
Параметр	Описание
Алгоритм аутентификации	Выберите алгоритм аутентификации.
Уровень привилегий	Выберите уровень привилегий.
Имя пользователя	Имя пользователя для аутентификации. Можно использовать пользовательские макросы.
Пароль	Пароль для аутентификации. Можно использовать пользовательские макросы.

Вкладка **Теги** позволяет определить теги на уровне узла сети. Все проблемы этого узла сети будут помечены введенными здесь значениями.



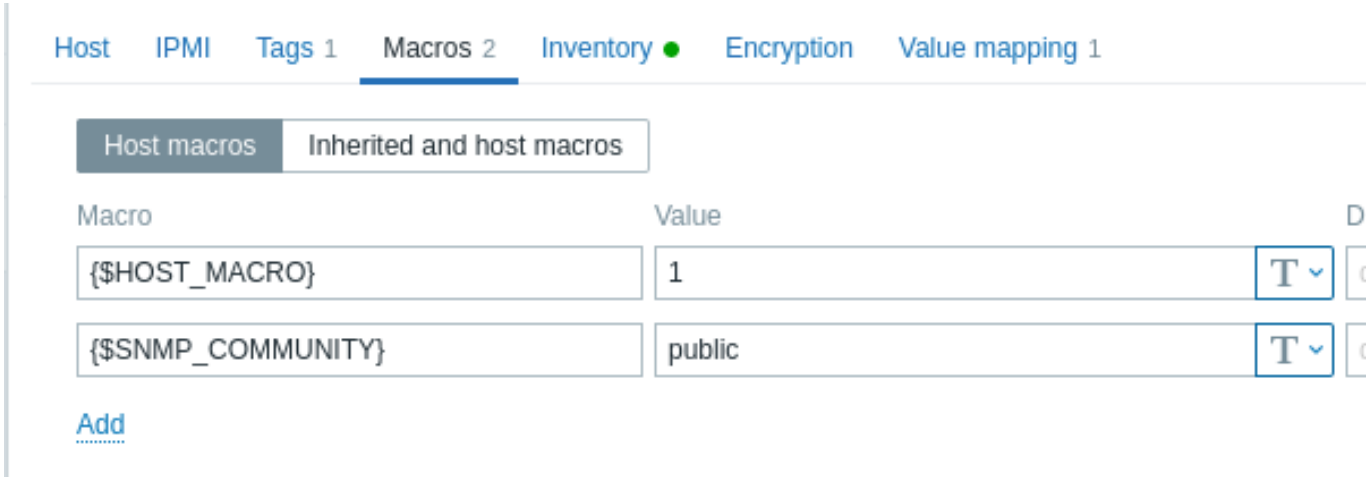
В тегах поддерживаются пользовательские макросы, макросы {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}.

Также здесь можно просматривать теги на уровне шаблона, если выбрать параметр *Унаследованные теги и теги узла сети*. Здесь отображаются все определенные пользовательские теги узла сети, а также их источник.

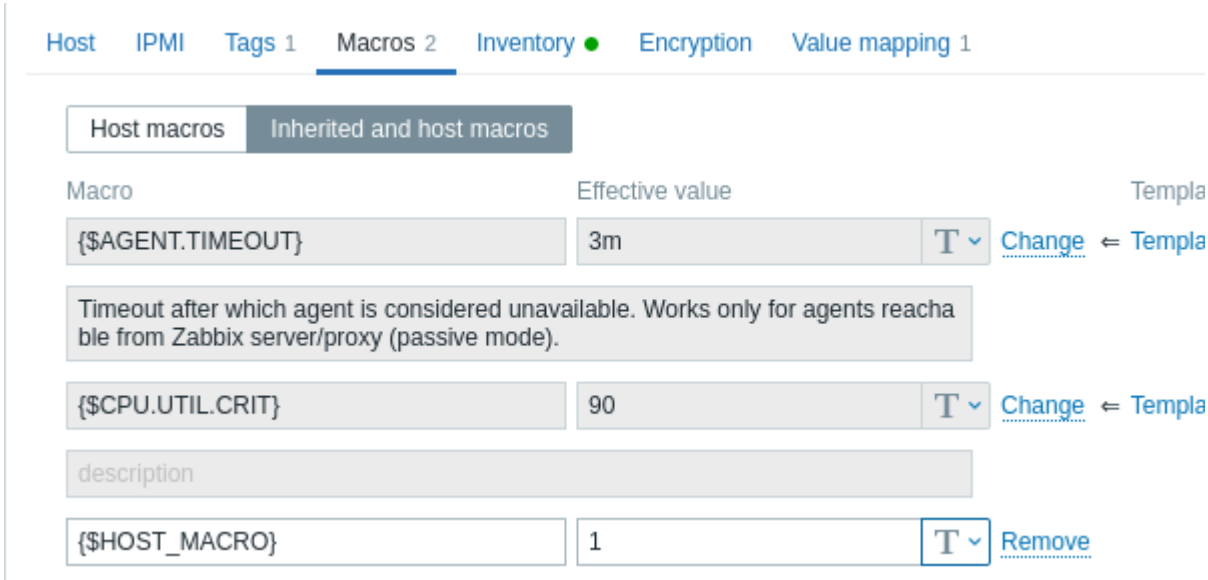


Для удобства предоставлены ссылки на соответствующие шаблоны.

Вкладка **Макросы** позволяет определить пользовательские макросы на уровне узла сети в виде пар «имя — значение». Обратите внимание, что значения макросов можно хранить как обычный текст, секретный текст или секрет Vault. Также можно добавить описание.

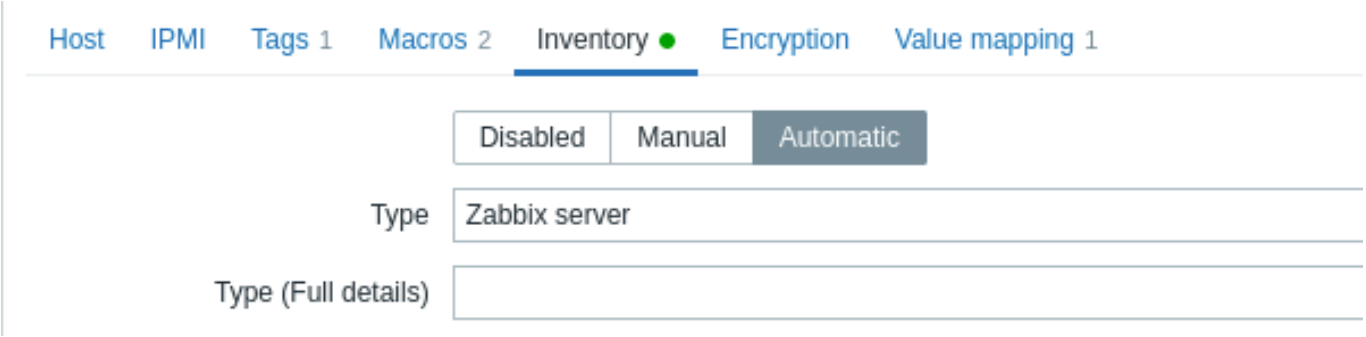


Также здесь можно просматривать пользовательские макросы на уровне шаблона и глобальные пользовательские макросы, если выбрать параметр *Унаследованные макросы и макросы узла сети*. Здесь отображаются все определенные пользовательские макросы узла сети, значения, в которые они преобразуются, а также их источник.



Для удобства предоставлены ссылки на соответствующие шаблоны и глобальную конфигурацию макросов. Также можно редактировать макрос шаблона или глобальный макрос на уровне узла сети, фактически создавая копию макроса на узле сети.

Вкладка **Инвентаризация** позволяет вручную ввести сведения об инвентаризации узла сети. Также можно включить заполнение инвентаризации в *Автоматическом* режиме или отключить заполнение инвентаризации для этого узла сети.



Если инвентаризация включена (вручную или автоматически), рядом с именем вкладки отображается зеленая точка.

Шифрование

Вкладка **Шифрование** позволяет требовать зашифрованные соединения с узлом сети.

Parameter	Description
Connections to host	Как сервер или прокси Zabbix подключается к агенту Zabbix на узле сети: без шифрования (по умолчанию), с использованием PSK (предварительно разделяемого ключа) или сертификата.
Connections from host	Выберите, какие типы соединений разрешены с узла сети (то есть от агента Zabbix и Zabbix sender). Можно выбрать несколько типов соединений одновременно (это полезно для тестирования и переключения на другой тип соединения). По умолчанию — ”No encryption”.
Issuer	Разрешенный издатель сертификата. Сначала сертификат проверяется с помощью CA (центра сертификации). Если он действителен и подписан CA, поле *Issuer* можно использовать для дополнительного ограничения разрешенного CA. Это поле предназначено для использования, если ваша установка Zabbix использует сертификаты от нескольких CA. Если это поле пустое, принимается любой CA.
Subject	Разрешенный субъект сертификата. Сначала сертификат проверяется с помощью CA. Если он действителен и подписан CA, поле *Subject* можно использовать, чтобы разрешить только одно значение строки *Subject*. Если это поле пустое, принимается любой действительный сертификат, подписанный настроенным CA.
PSK identity	Строка идентификатора предварительно разделяемого ключа. Не помещайте конфиденциальную информацию в PSK identity, так как она передается по сети без шифрования, чтобы сообщить получателю, какой PSK использовать.
PSK	Предварительно разделяемый ключ (hex-строка). Максимальная длина: 512 шестнадцатеричных цифр (PSK длиной 256 байт), если Zabbix использует библиотеку GnuTLS или OpenSSL, 64 шестнадцатеричные цифры (PSK длиной 32 байта), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Сопоставление значений

Вкладка **Сопоставление значений** позволяет настроить удобное для пользователя представление данных элемента данных в сопоставлениях значений.

Настройка группы узлов сети

Обзор

Группы узлов сети используются для логического объединения узлов сети и назначения им прав доступа пользователей.

Каждому узлу сети должна быть назначена как минимум одна группа узлов сети. Узел сети может принадлежать нескольким группам узлов сети, а каждая группа узлов сети может содержать несколько узлов сети.

Обратите внимание, что в Zabbix все права доступа основаны на группах: группах пользователей, группах узлов сети и группах шаблонов. Таким образом, даже если одному пользователю нужен доступ к одному узлу сети, он предоставляется путем добавления пользователя в группу пользователей и назначения этой группе пользователей прав доступа к группе узлов сети, содержащей этот узел сети.

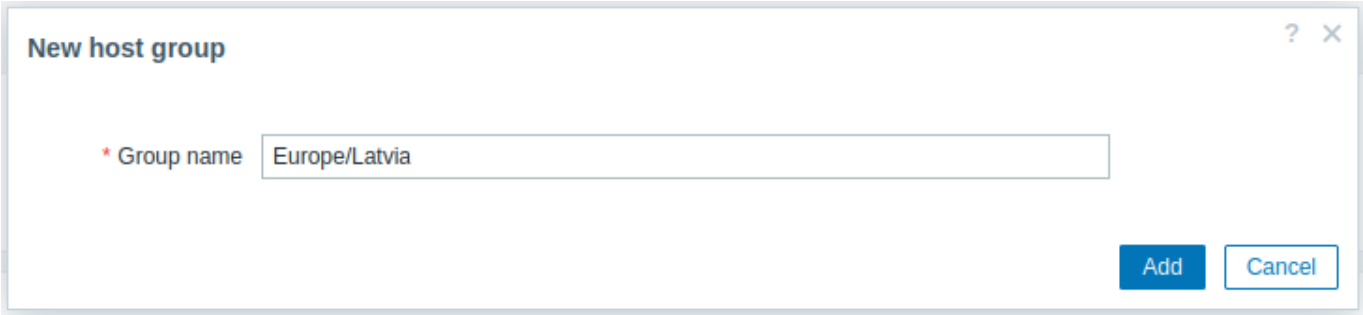
Конфигурация

> **Attention:**
> Только пользователи с ролью Super admin могут создавать группы узлов сети.

В веб-интерфейсе Zabbix есть два способа создания группы узлов сети.

Вариант 1:

1. Перейдите в *Сбор данных* > *Группы узлов сети*.
2. Нажмите *Создать группу узлов сети* в правом верхнем углу экрана.
3. Введите имя группы в форме.



Вариант 2: при настройке узла сети введите несуществующее имя группы в поле *Группы узлов сети*.

После создания группы узлов сети вы можете нажать на имя группы в списке в разделе *Сбор данных* > *Группы узлов сети*, чтобы изменить имя группы, клонировать группу или удалить ее.

При удалении группы узлов сети удаляется только логическая группа, а не узлы сети в этой группе. Невозможно удалить группу узлов сети, если она является единственной группой для какого-либо существующего узла сети.

Создание подгрупп узлов сети

Подгруппа узлов сети (или вложенная группа узлов сети) является дочерней по отношению к родительской группе узлов сети, которая ее содержит.

Подгруппа создается с помощью символа косой черты '/' в поле ввода имени группы, чтобы обозначить ее отношение к родительской группе или группам. Например:

- при вводе `Europe/Latvia` создается подгруппа `Europe/Latvia` родительской группы `Europe`.
- при вводе `Europe/Latvia/Riga/Zabbix servers` создается соответствующая подгруппа во вложенных родительских группах `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

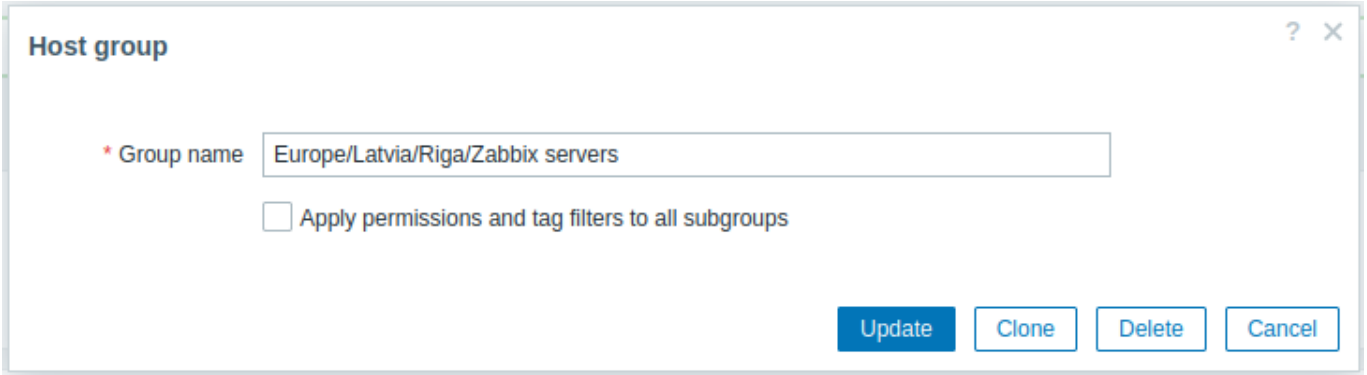
При создании подгруппы не допускается использование начальных или конечных косых черт, а также нескольких косых черт подряд. Экранирование символа '/' не поддерживается.

Перед созданием подгруппы не требуется предварительно создавать родительскую группу или группы узлов сети. Вы можете начать либо с создания подгруппы, например `Europe/Latvia`, либо с создания любой родительской группы или групп узлов сети (в нашем примере — `Europe`). Если вы начнете с создания подгруппы, родительская группа или группы узлов сети **не** будут созданы автоматически.

Права доступа к группам узлов сети

- При создании подгруппы в существующей родительской группе узлов сети (например, при создании `Europe/Latvia`, когда `Europe` уже существует), групповые пользователи наследуют права доступа к подгруппе от родительской группы.
- При создании родительской группы узлов сети для уже существующей подгруппы (например, при создании `Europe`, когда `Europe/Latvia` уже существует), права доступа к родительской группе не задаются.

При редактировании любой группы узлов сети можно также задать дополнительную опцию *Применить права доступа и фильтры по тегам ко всем подгруппам*.



Если установить этот флажок и нажать *Обновить*, будут применены те же уровни прав доступа и фильтры по тегам ко всем текущим и будущим подгруппам редактируемой группы узлов сети.

Таким образом, если каким-либо группам пользователей были назначены разные права доступа к подгруппам редактируемой группы узлов сети, установка флажка предоставит всем текущим и будущим подгруппам те же права доступа пользователей и права доступа на основе тегов, что и у редактируемой группы.

Обратите внимание, что эта опция не сохраняется в базе данных и переопределяет существующие права доступа. Любые изменения, внесенные с помощью этой опции, можно отменить только вручную.

Инвентаризация

Обзор

Вы можете хранить инвентарные данные сетевых устройств в Zabbix.

В веб-интерфейсе Zabbix существует специальное меню *Инвентаризация (Inventory)*. Однако, поначалу вы не увидите там никаких данных, и это не то место, где вы вводите данные. Построение инвентарных данных выполняется вручную при настройке узла сети или автоматически с использованием каких-либо опций автоматического заполнения.

Построение инвентарных данных

Ручной режим

При настройке узла сети на вкладке *Инвентаризация* можно указать такие сведения, как тип устройства, серийный номер, местоположение, ответственное лицо, URL-адреса и т. д. — данные, которые будут использоваться для заполнения информации об инвентаризации.

Если в информации об инвентаризации узла сети указан URL-адрес, начинающийся с «http» или «https», в разделе *Инвентаризация* он будет отображаться как активная ссылка.

Автоматический режим

Инвентаризация узла сети также может выполняться автоматически. Для этого при настройке узла сети в поле режима инвентаризации на вкладке *Инвентаризация* необходимо выбрать значение *Автоматический*.

Затем можно настроить элементы данных узла сети, чтобы заполнять любое поле инвентаризации узла сети их значением, указав целевое поле с помощью соответствующего атрибута (называемого *Элемент данных заполнит поле инвентаризации узла сети*) в настройках элемента данных.

Элементы данных, особенно полезные для автоматического сбора данных инвентаризации:

- system.hw.chassis[full|type|vendor|model|serial] - по умолчанию используется [full], требуются права root
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - по умолчанию используется [all,full]
- system.hw.devices[pci|usb] - по умолчанию используется [pci]
- system.hw.macaddr[interface,short|full] - по умолчанию используется [all,full], interface представляет собой регулярное выражение
- system.sw.arch
- system.sw.os[name|short|full] - по умолчанию используется [name]
- system.sw.packages[regexp,manager,short|full] - по умолчанию используется [all,all,full].

Выбор режима инвентаризации

Режим инвентаризации можно выбрать в форме настройки узла сети.

Режим инвентаризации по умолчанию для новых узлов сети выбирается на основе настройки *Режим инвентаризации узла сети по умолчанию* в разделе *Администрирование* > *Общие* > *Прочее*.

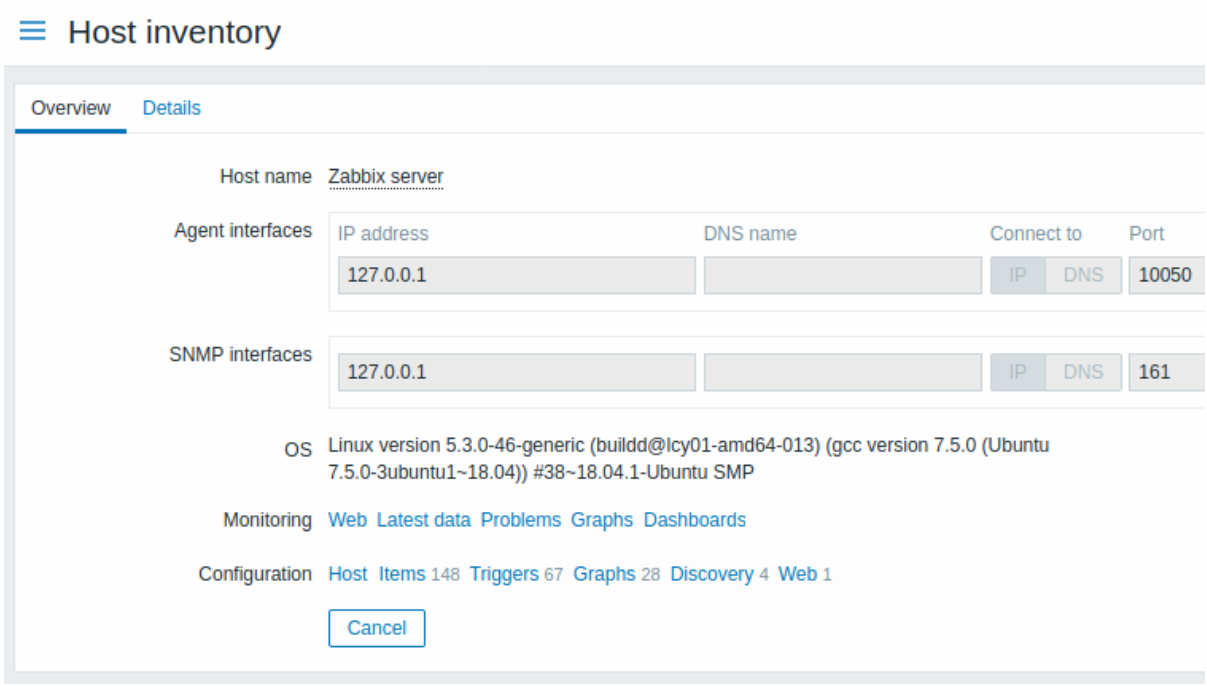
Для узлов сети, добавленных в результате обнаружения сети или действий автоматической регистрации, можно определить операцию *Установить режим инвентаризации узла сети*, выбрав ручной или автоматический режим. Эта операция переопределяет настройку *Режим инвентаризации узла сети по умолчанию*.

Обзор инвентаризации

Подробные сведения обо всех существующих данных инвентаризации доступны в меню *Инвентаризация*.

В *Инвентаризация* > *Обзор* можно получить количество узлов сети по различным полям инвентаризации.

В *Инвентаризация* > *Узлы сети* можно увидеть все узлы сети, для которых есть информация об инвентаризации. При нажатии на имя узла сети откроются подробные сведения об инвентаризации в форме.



Вкладка **Обзор** показывает:

Parameter	Description
Host name	Имя узла сети. При нажатии на имя открывается меню со скриптами, определенными для узла сети. Имя узла сети отображается с оранжевым значком, если узел сети находится на обслуживании.
Visible name	Отображаемое имя узла сети (если задано).
Host (Agent, SNMP, JMX, IPMI) interfaces	Этот блок содержит сведения об интерфейсах, настроенных для узла сети.
OS	Поле инвентаризации операционной системы узла сети (если задано).
Hardware	Поле инвентаризации аппаратного обеспечения узла сети (если задано).
Software	Поле инвентаризации программного обеспечения узла сети (если задано).
Description	Описание узла сети.
Monitoring	Ссылки на разделы мониторинга с данными для этого узла сети: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Ссылки на разделы конфигурации для этого узла сети: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Количество настроенных сущностей указано после каждой ссылки.

Вкладка **Details** показывает все заполненные поля инвентаризации (не пустые).

Макросы инвентаризации

Для использования в уведомлениях доступны макросы инвентаризации узла сети `{INVENTORY.*}`, например:

”На сервере в {INVENTORY.LOCATION1} возникла проблема, ответственное лицо — {INVENTORY.CONTACT1}, номер телефона {INVENTORY.POC.PRIMARY.PHONE.A1}.”

Подробнее см. страницу Поддерживаемые макросы.

Массовое обновление

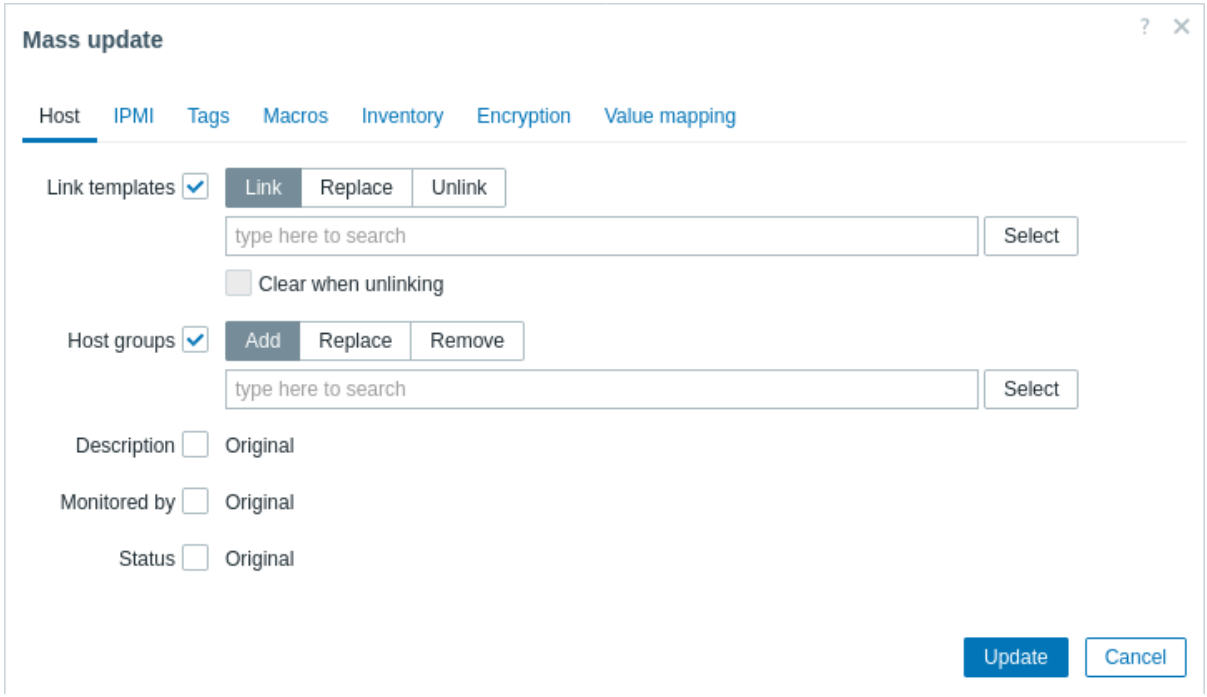
Обзор

Порой вы можете захотеть изменить какие-нибудь атрибуты для нескольких узлов сети одновременно. Вместо того, чтобы открывать каждый узел сети для редактирования, вы можете воспользоваться функцией массового обновления.

Массовое обновление

Чтобы выполнить массовое обновление нескольких узлов сети, выполните следующие действия:

- Установите флажки перед узлами сети, которые требуется обновить, в списке узлов сети
- Нажмите *Массовое обновление* под списком
- Перейдите на вкладку с необходимыми атрибутами (*Узел сети*, *IPMI*, *Теги*, *Макросы*, *Инвентаризация*, *Шифрование* или *Сопоставление значений*)
- Установите флажки нужных атрибутов для обновления и введите для них новое значение



При выборе соответствующей кнопки для обновления связей с **шаблонами** доступны следующие параметры:

- *Связать* - укажите дополнительные шаблоны для связывания
- *Заменить* - укажите шаблоны для связывания, отменив связь с шаблонами, которые ранее были связаны с узлами сети
- *Отменить связь* - укажите шаблоны, связь с которыми требуется отменить

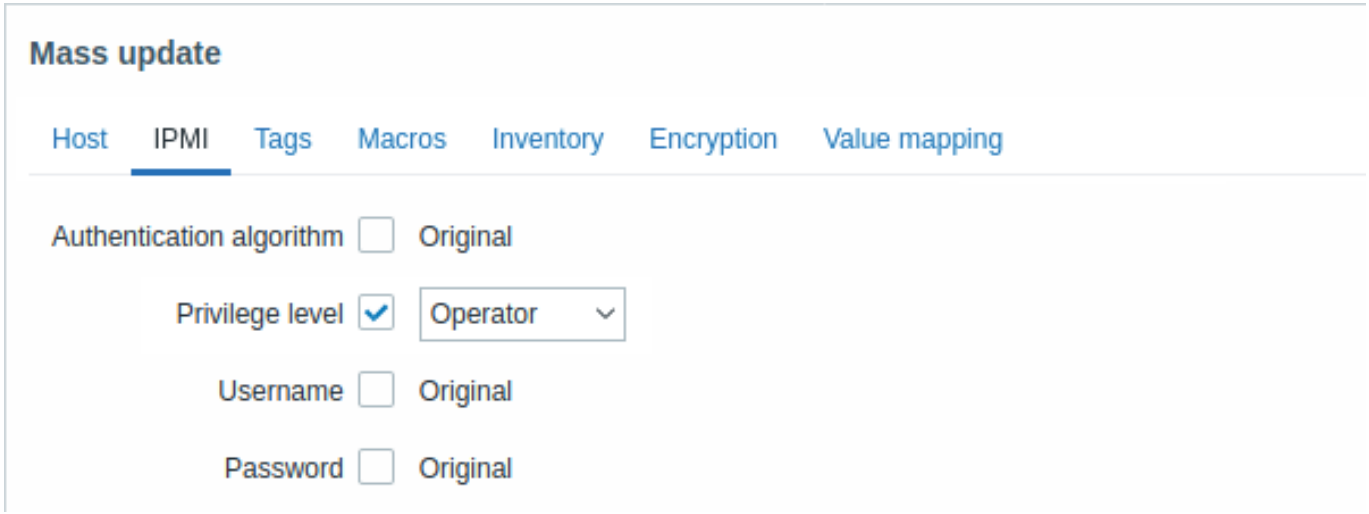
Чтобы указать шаблоны для связывания или отмены связи, начните вводить имя шаблона в поле автозаполнения, пока не появится раскрывающийся список с подходящими шаблонами. Прокрутите список вниз и выберите нужный шаблон.

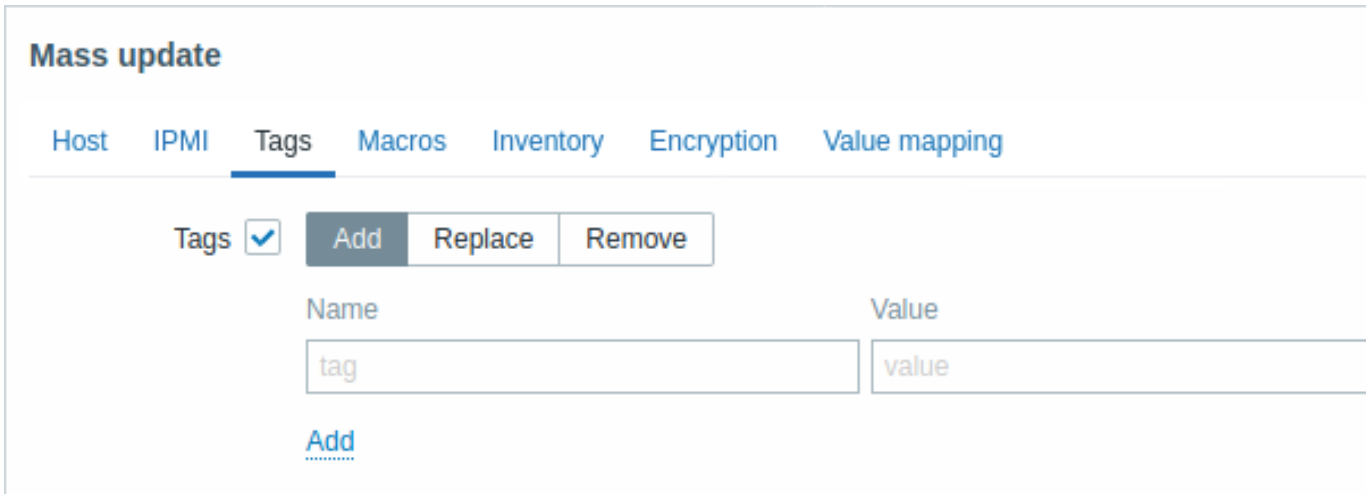
Параметр *Очистить при отмене связи* позволяет не только отменить связь с ранее связанными шаблонами, но и удалить все унаследованные от них сущности (элементы данных, триггеры, графики и т. д.).

При выборе соответствующей кнопки для обновления **групп узлов сети** доступны следующие параметры:

- *Добавить* - укажите дополнительные группы узлов сети из существующих или введите совершенно новые группы узлов сети для узлов сети;
- *Заменить* - удалите узел сети из всех существующих групп узлов сети и замените их группой или группами, указанными в этом поле (существующими или новыми группами узлов сети);
- *Удалить* - удалите определенные группы узлов сети из узлов сети.

Эти поля поддерживают автозаполнение: при вводе отображается раскрывающийся список подходящих групп узлов сети. Если группа узлов сети новая, она также отображается в раскрывающемся списке и помечается строкой *(новая)*. Прокрутите список вниз и выберите нужную группу.

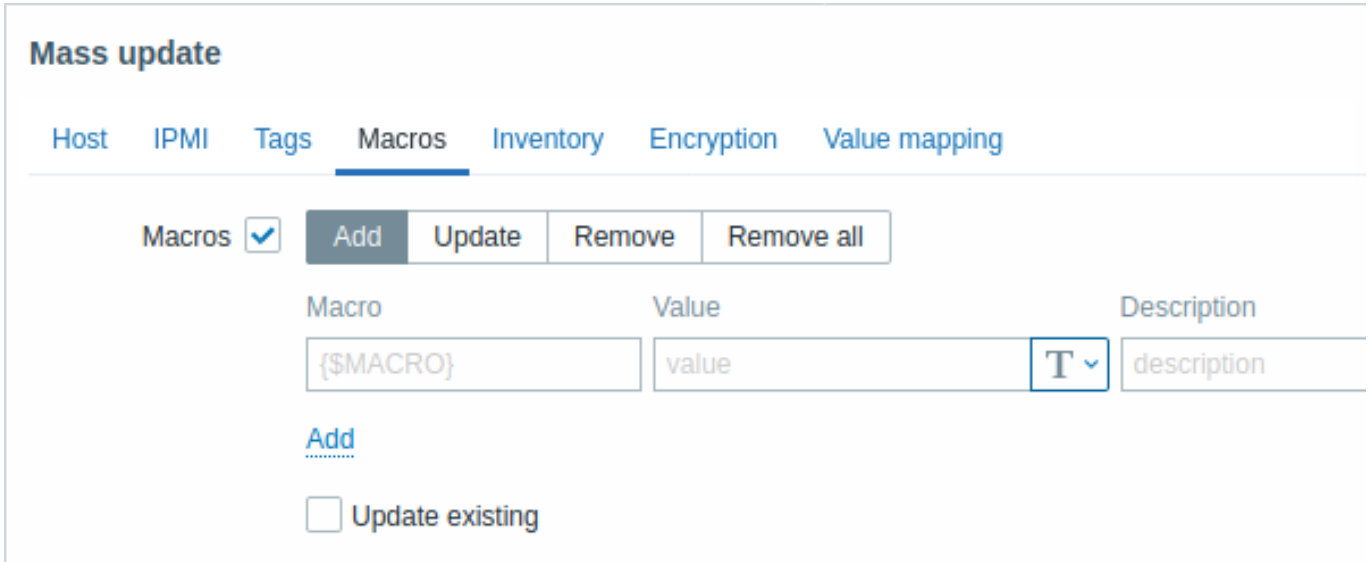




При выборе соответствующей кнопки для обновления тегов доступны следующие параметры:

- *Добавить* - добавьте новые теги для узлов сети;
- *Заменить* - удалите все существующие теги узла сети и замените их тегом или тегами, указанными ниже;
- *Удалить* - удалите указанные теги из узлов сети.

В тегах поддерживаются пользовательские макросы, макросы `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` и `{HOST.ID}`. Обратите внимание, что теги с одинаковым именем, но разными значениями, не считаются «дубликатами» и могут быть добавлены к одному узлу сети.

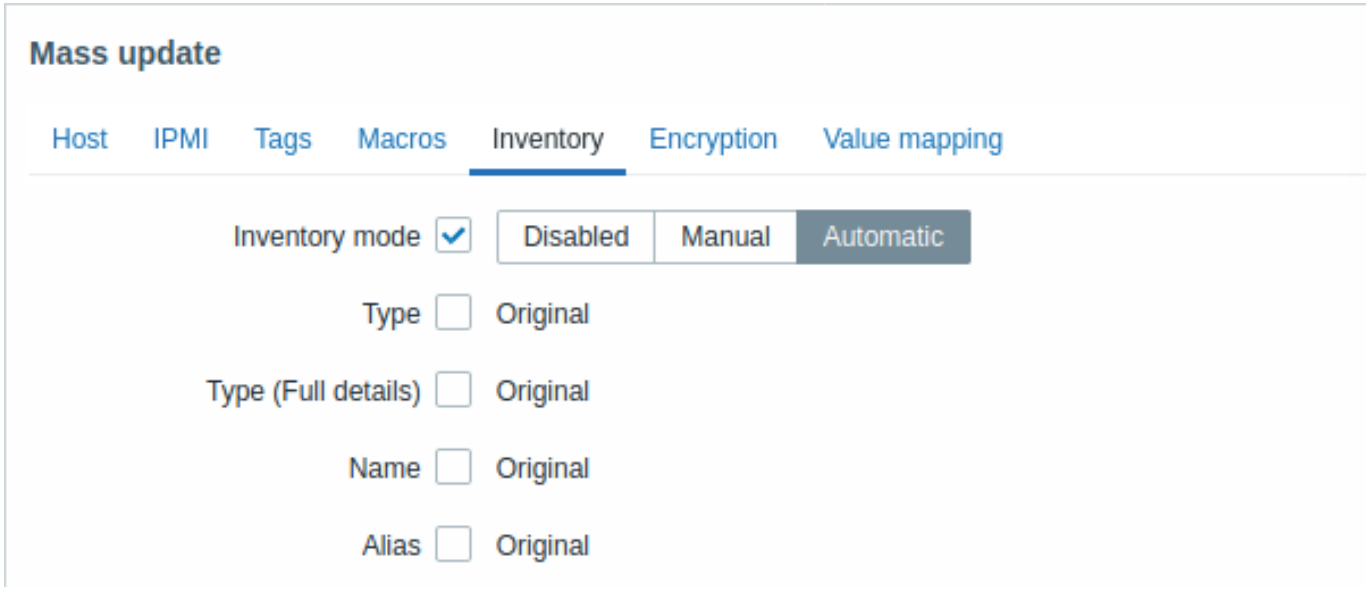


При выборе соответствующей кнопки для обновления макросов доступны следующие параметры:

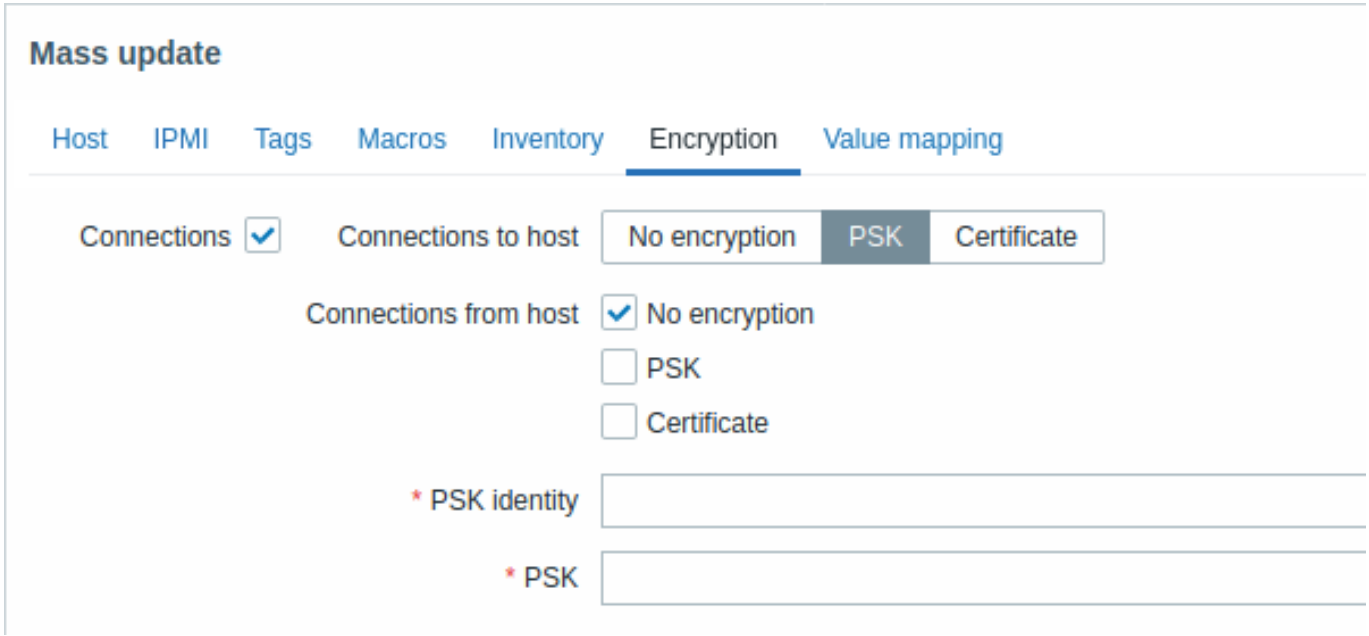
- *Добавить* - укажите дополнительные пользовательские макросы для узлов сети. Если установлен флажок *Обновить существующие*, значения, тип и описание указанного имени макроса будут обновлены. Если флажок не установлен

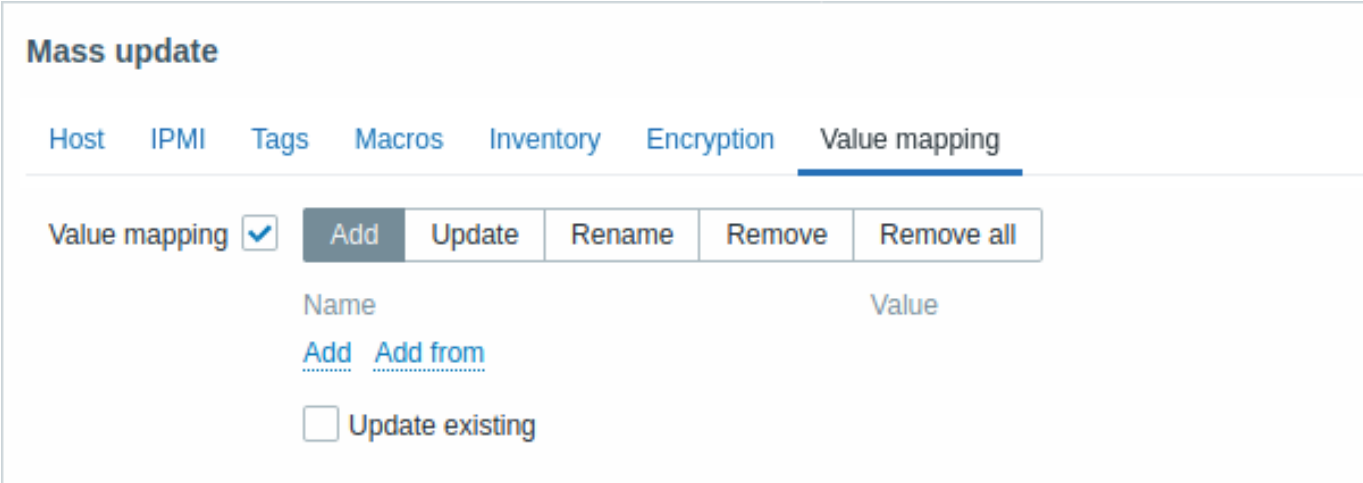
и макрос с таким именем уже существует на узле или узлах сети, он не будет обновлен.

- *Обновить* - замените значения, типы и описания макросов, указанных в этом списке. Если установлен флажок *Добавить отсутствующие*, макрос, которого ранее не было на узле сети, будет добавлен как новый макрос. Если флажок не установлен, будут обновлены только макросы, которые уже существуют на узле сети.
- *Удалить* - удалите указанные макросы из узлов сети. Если установлен флажок *Кроме выбранных*, будут удалены все макросы, кроме указанных в списке. Если флажок не установлен, будут удалены только макросы, указанные в списке.
- *Удалить все* - удалите все пользовательские макросы из узлов сети. Если флажок *Я подтверждаю удаление всех макросов* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления всех макросов.



Чтобы иметь возможность массово обновлять поля инвентаризации, для параметра *Режим инвентаризации* должно быть установлено значение «Вручную» или «Автоматически».





Для обновления сопоставления значений доступны кнопки со следующими параметрами:

- *Добавить* - добавьте сопоставления значений к узлам сети. Если установить флажок *Обновить существующие*, все свойства сопоставления значений с этим именем будут обновлены. В противном случае, если сопоставление значений с таким именем уже существует, оно не будет обновлено.
- *Обновить* - обновите существующие сопоставления значений. Если установить флажок *Добавить отсутствующие*, сопоставление значений, которого ранее не было на узле сети, будет добавлено как новое сопоставление значений. В противном случае будут обновлены только сопоставления значений, которые уже существуют на узле сети.
- *Переименовать* - задайте новое имя для существующего сопоставления значений
- *Удалить* - удалите указанные сопоставления значений из узлов сети. Если установить флажок *Кроме выбранных*, все сопоставления значений будут удалены **за исключением** указанных.
- *Удалить все* - удалите все сопоставления значений из узлов сети. Если флажок *Я подтверждаю удаление всех сопоставлений значений* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления.

После внесения всех необходимых изменений нажмите *Обновить*. Атрибуты будут соответствующим образом обновлены для всех выбранных узлов сети.

Элементы данных

Обзор

Элемент данных является отдельной метрикой.

Элементы данных используются для сбора данных.

Когда вы настроили узел сети, для получения реальных данных вам нужно добавить элементы данных. Один из способов быстро добавить большое количество элементов данных, — это присоединить к узлу сети один из предопределённых шаблонов. Однако, для оптимизации производительности системы, вам может потребоваться выполнить тонкую настройку шаблонов, чтобы у вас было ровно столько элементов данных и они собирались бы настолько часто, насколько это действительно необходимо.

Чтобы указать, какого рода данные будут собираться с узла сети, используйте ключ элемента данных. Например, элемент данных с именем ключа **system.cpu.load** будет собирать данные о загрузке процессора, тогда как элемент данных с именем ключа **net.if.in** будет собирать информацию о входящем трафике.

Дополнительные параметры могут быть указаны в квадратных скобках после имени ключа. Например, system.cpu.load**[avg5]** будет возвращать загрузку процессора за последние 5 минут, а net.if.in**[eth0]** покажет входящий трафик на интерфейсе eth0.

Note:
Для просмотра всех поддерживаемых типов элементов данных и ключей элементов данных смотрите отдельные разделы по типам элементов данных.

Перейти к созданию и настройке элемента данных.

Создание элемента данных

Обзор

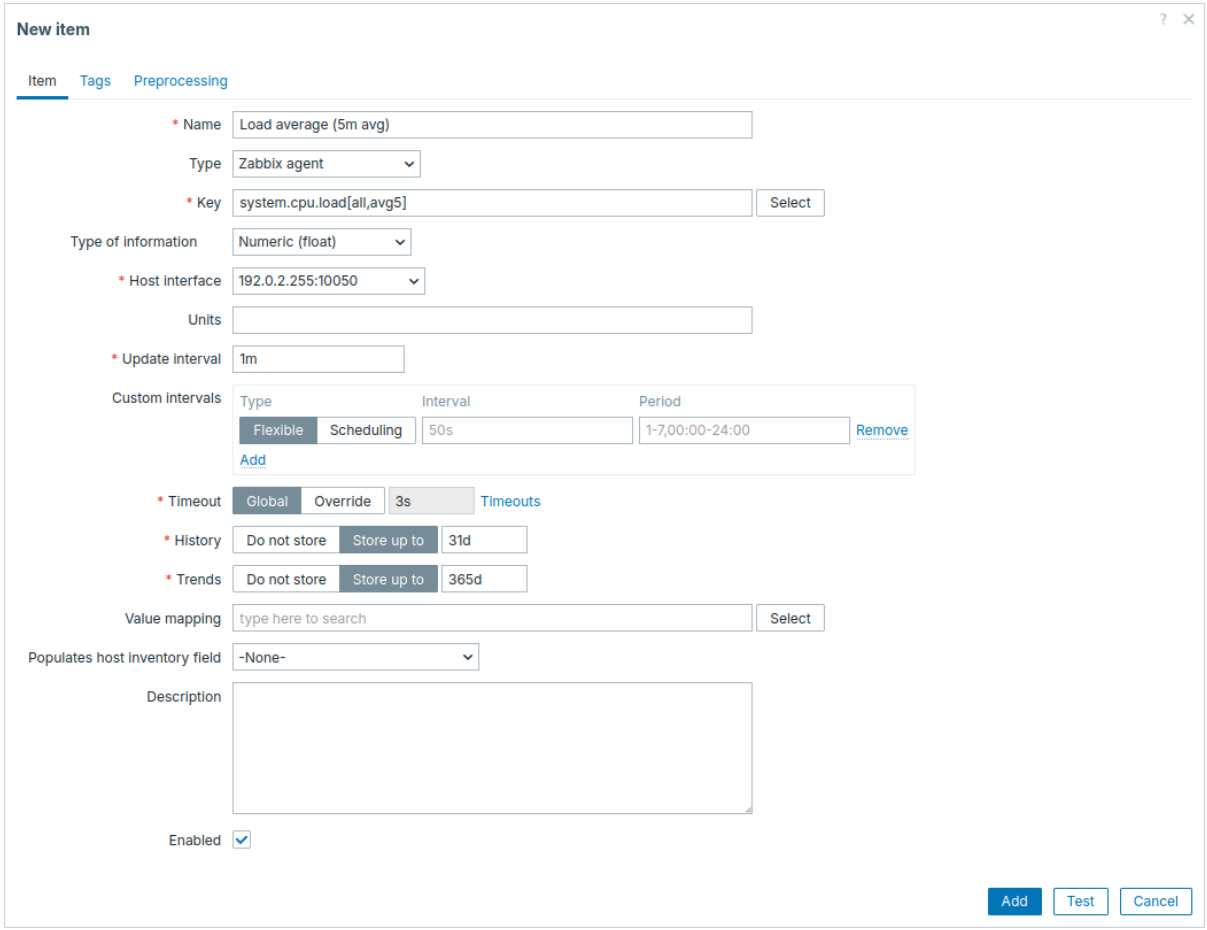
Чтобы создать элемент данных в веб-интерфейсе Zabbix, выполните следующие действия:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Нажмите *Элементы данных* в строке узла сети.
3. Нажмите *Создать элемент данных* в правом верхнем углу экрана.
4. Укажите параметры элемента данных в форме.

Также можно создать элемент данных, открыв существующий элемент данных, нажав кнопку *Клонировать* и сохранив его под другим именем.

Настройка

Вкладка **Элемент данных** содержит общие атрибуты элемента данных.



Все обязательные поля отмечены красной звездочкой.

Параметр	Описание
Имя	Имя элемента данных. Поддерживаются пользовательские макросы.
Тип	Тип элемента данных. См. отдельные разделы о типах элементов данных.
Ключ	Ключ элемента данных (до 2048 символов). Поддерживаемые ключи элементов данных приведены в отдельных разделах о типах элементов данных. Ключ должен быть уникальным в пределах одного узла сети. Если тип ключа — «Zabbix agent», «Zabbix agent (active)» или «Simple check», значение ключа должно поддерживаться агентом Zabbix или сервером Zabbix. См. также правильный формат ключа.

Параметр	Описание
Тип информации	Тип данных, используемый для проверки значения элемента данных и его хранения в базе данных, в том числе после выполнения любых преобразований: **Числовой (без знака)** — 64-разрядное целое число без знака; **Числовой (с плавающей точкой)** — 64-разрядное число с плавающей точкой; **Символ** — короткий текст; **Журнал** — длинный текст с параметрами журнала или без них (временная метка, источник, важность, идентификатор события журнала); **Текст** — длинный текст; **Двоичный** — двоичное число (поддерживается только для зависимых элементов данных); **JSON** — структурированные данные JSON, хранящиеся в базе данных в исходном формате (не поддерживается для вычисляемых элементов данных). Элементы данных, возвращающие значения с двоичным типом данных или типом JSON, не поддерживаются в формулах вычисляемых элементов данных или выражениях триггеров. Ограничения хранения для каждого типа данных и серверной части базы данных см. в разделе Ограничения данных элементов данных. Для элементов данных, возвращающих данные в одном определенном формате, соответствующий тип данных выбирается автоматически — например, `system.cpu.load` и `system.cpu.util` возвращают 64-разрядное число с плавающей точкой, поэтому автоматически выбирается тип *Числовой (с плавающей точкой)*.
Интерфейс узла сети	Выберите интерфейс узла сети. Это поле доступно при редактировании элемента данных на уровне узла сети.
Единицы измерения	Если задан символ единицы измерения, Zabbix выполняет постобработку полученного значения элемента данных и отображает его с указанным суффиксом единицы измерения. Поддерживаемые символы единиц измерения со специальным форматированием (и примеры: полученное значение элемента данных → отображаемое значение): **B** — байты (1024 → 1 KB) **Bps** — байты в секунду (1024 → 1 KBps) **s** — секунды, отображаемые с использованием до трех крупнейших ненулевых единиц времени (881764 → 10d 4h 56m) **uptime** — прошедшее время в формате чч:мм:сс или N дней, чч:мм:сс (881764 → 10 days, 04:56:04) **unixtime** — временная метка Unix в формате гггг.мм.дд чч:мм:сс (881764 → 1970-01-11 04:56:04 AM); для правильного форматирования полученное значение элемента данных должно иметь тип *Числовой (без знака)*. Для других единиц измерения (например, Гц, Вт и т. д.), если полученное значение превышает 1000, оно делится на 1000 и отображается с соответствующим префиксом (5000 → 5 KHz, 881764 → 881.76 KW). Если перед символом единицы измерения указан !, преобразование единиц измерения и добавление префикса отключаются (1024 !B → 1024 B, 61 !s → 61 s). Дополнительные примеры и сведения о символах единиц измерения и преобразовании единиц см. в разделе Суффиксы значений элементов данных.
Интервал обновления	Получать новое значение для этого элемента данных каждые N секунд. Максимально допустимый интервал обновления — 86400 секунд (1 день). Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. Один макрос должен занимать всё поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются. *Примечание*: Интервал обновления можно установить равным «0» только при наличии пользовательских интервалов с ненулевым значением. Если задано значение «0» и существует пользовательский интервал (гибкий или запланированный) с ненулевым значением, опрос элемента данных будет выполняться в течение пользовательского интервала. *Обратите внимание*, что первый опрос элемента данных после его активации или изменения интервала обновления может произойти раньше заданного значения. Новые элементы данных будут проверены в течение 60 секунд после создания, если для них не задан интервал обновления «Расписание» или «Гибкий» и параметр *Интервал обновления* не установлен в значение 0. Существующий пассивный элемент данных можно немедленно опросить, нажав кнопку *Выполнить сейчас* button.

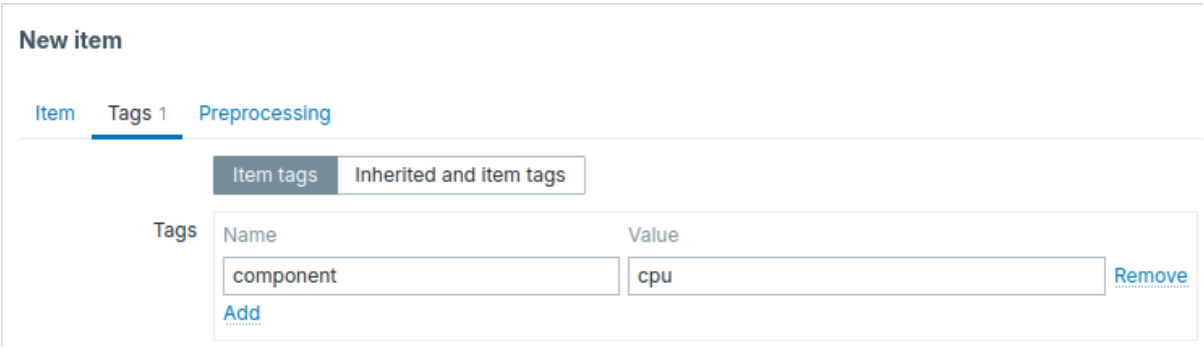
Параметр	Описание
Пользовательские интервалы	Можно создать пользовательские правила проверки элемента данных: **Гибкий** — создать исключение для *Интервала обновления* (интервал с другой частотой). **Расписание** — создать пользовательское расписание опроса. Подробную информацию см. в разделе Пользовательские интервалы. В поле *Интервал* поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. Один макрос должен занимать всё поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются.
Тайм-аут	Задайте тайм-аут проверки элемента данных (доступно для поддерживаемых типов элементов данных). Выберите вариант тайм-аута: **Глобальный** — используется тайм-аут прокси/глобальный тайм-аут (отображается в неактивном поле *Тайм-аут*). **Переопределить** — используется пользовательский тайм-аут (задается в поле *Тайм-аут*; допустимый диапазон: 1–600 с). Поддерживаются суффиксы времени, например 30s, 1m, и пользовательские макросы. Нажатие ссылки *Тайм-ауты* позволяет настроить тайм-ауты прокси или глобальные тайм-ауты (если прокси не используется). Обратите внимание, что ссылка *Тайм-ауты* видна только пользователям типа *Суперадминистратор*, имеющим разрешения на разделы веб-интерфейса *Администрирование* > *Общие* или *Администрирование* > *Прокси*.
История	Выберите один из вариантов: **Не хранить** — история элемента данных не хранится. Полезно для основных элементов данных, если историю необходимо сохранять только для зависимых элементов данных. Этот параметр нельзя переопределить глобальными настройками очистки. **Хранить до** — укажите срок хранения подробной истории в базе данных (от 1 часа до 25 лет). Более старые данные будут удалены очисткой. Хранится в секундах. Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы. Значение *Хранить до* можно глобально переопределить в разделе *Администрирование* > *Очистка*. Если задана глобальная настройка переопределения, отображается оранжевый информационный значок. При наведении указателя мыши отображается предупреждение, например: *Переопределено глобальными настройками очистки (1d)*. Рекомендуется хранить записанные значения минимально возможное время, чтобы уменьшить размер истории значений в базе данных. Вместо длительного хранения истории значений можно дольше хранить данные трендов. См. также раздел История и тренды.
Тренды	Выберите один из вариантов: **Не хранить** — тренды не хранятся. Этот параметр нельзя переопределить глобальными настройками очистки. **Хранить до** — укажите срок хранения агрегированной истории (минимальное, максимальное и среднее значение за час, количество) в базе данных (от 1 дня до 25 лет). Более старые данные будут удалены очисткой. Хранится в секундах. Поддерживаются суффиксы времени, например 24h, 1d. Поддерживаются пользовательские макросы. Значение *Хранить до* можно глобально переопределить в разделе *Администрирование* > *Очистка*. Если задана глобальная настройка переопределения, отображается оранжевый информационный значок. При наведении указателя мыши отображается предупреждение, например: *Переопределено глобальными настройками очистки (7d)*. *Примечание:* Хранение трендов недоступно для нечисловых данных — символов, журналов и текста. См. также раздел История и тренды.
Сопоставление значений	Примените сопоставление значений к этому элементу данных. Сопоставление значений не изменяет полученные значения, а используется только для отображения данных. Работает с элементами данных типов *Числовой (без знака)*, *Числовой (с плавающей точкой)* и *Символ*. Например, «Состояния служб Windows».

Параметр	Описание
Формат времени журнала	Доступно только для элементов данных типа **Журнал**. Поддерживаемые заполнители: **y**: *Год (1970–2038)*. **M**: *Месяц (01–12)*. **d**: *День (01–31)*. **h**: *Час (00–23)*. **m**: *Минута (00–59)*. **s**: *Секунда (00–59)*. Если поле оставить пустым, временная метка будет установлена в значение 0 времени Unix, соответствующее 1 января 1970 года. Например, рассмотрим следующую строку из файла журнала агента Zabbix: ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” Она начинается с шести позиций символов для PID, за которыми следуют дата, время и остальная часть сообщения. Формат времени журнала для этой строки будет иметь вид ”ppppp:yyyyMMdd:hhmmss”. Обратите внимание, что символы «p» и «:» являются заполнителями и могут быть любыми символами, кроме «yMdhms».
Заполняет поле инвентаризации узла сети	Можно выбрать поле инвентаризации узла сети, которое будет заполняться значением элемента данных. Это работает, если для узла сети включено автоматическое заполнение инвентаризации, а также для элементов данных типов *Числовой (без знака)*, *Числовой (с плавающей точкой)*, *Символ* и *Текст*.
Описание	Введите описание элемента данных. Поддерживаются пользовательские макросы.
Включен	Установите флажок, чтобы включить элемент данных и разрешить его обработку. При отключении элемента данных он немедленно удаляется из кэша истории (за исключением последнего значения, которое сохраняется для журналов).
Последние данные	Нажмите ссылку, чтобы просмотреть последние данные элемента данных. Ссылка доступна только при редактировании уже существующего элемента данных.

Note:
Поля, специфичные для типа элемента данных, описаны на соответствующих страницах.

Note:
При редактировании существующего элемента данных уровня шаблона на уровне узла сети ряд полей доступен только для чтения. Можно воспользоваться ссылкой в заголовке формы, перейти на уровень шаблона и отредактировать их там, учитывая, что изменения на уровне шаблона изменят элемент данных для всех узлов сети, с которыми связан шаблон.

Вкладка **Теги** позволяет определить теги уровня элемента данных.



Предобработка значений элемента данных

Вкладка **Предобработка** позволяет определить правила преобразования для полученных значений.

Тестирование элемента данных

Attention:
Чтобы выполнить тестирование элемента данных, убедитесь, что системное время на сервере и прокси синхронизировано. Если время на сервере отстает, при тестировании элемента данных может быть возвращено сообщение об ошибке ”The task has been expired.” Однако разные часовые пояса на сервере и прокси не повлияют на результат тестирования.

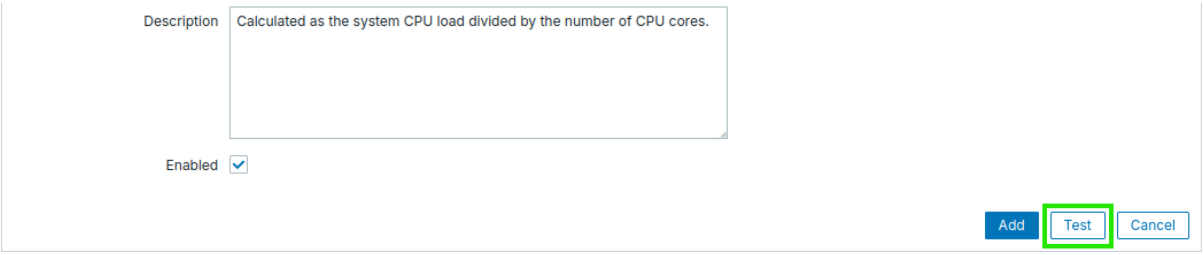
Можно протестировать элемент данных и, если он настроен правильно, получить в ответ реальное значение. Тестирование может выполняться даже до сохранения элемента данных.

Тестирование доступно для элементов данных узла сети и шаблона, прототипов элементов данных и правил обнаружения низкого уровня. Тестирование недоступно для активных элементов данных.

Тестирование элемента данных доступно для следующих пассивных типов элементов данных:

- Zabbix agent
- SNMP agent (v1, v2, v3)
- IPMI agent
- SSH checks
- Telnet checks
- JMX agent
- Simple checks (except `icmpping*`, `vmware.*` items)
- Zabbix internal
- Calculated items
- External checks
- Database monitor
- HTTP agent
- Script
- Browser

Чтобы протестировать элемент данных, нажмите кнопку *Test* в нижней части формы настройки элемента данных. Обратите внимание, что кнопка *Test* будет недоступна для элементов данных, которые нельзя протестировать (например, активные проверки, исключенные простые проверки).

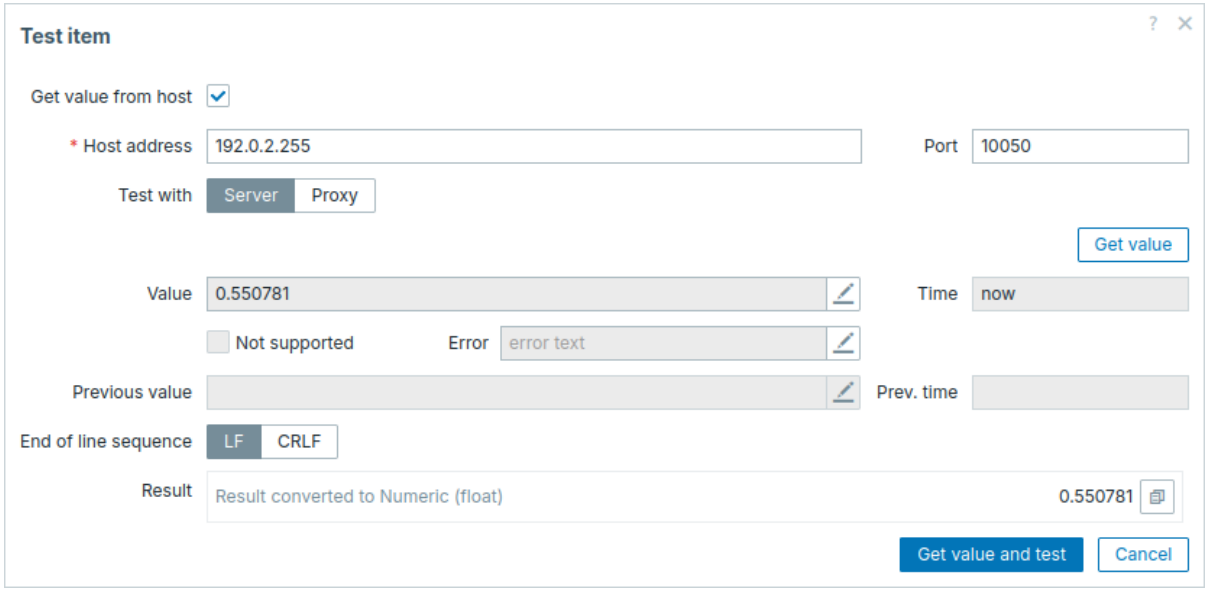


Форма тестирования элемента данных содержит поля для обязательных параметров узла сети (адрес узла сети, порт, тестирование с сервером/прокси (имя прокси)) и специфичные для элемента данные (например, SNMPv2 community или учетные данные безопасности SNMPv3). Эти поля учитывают контекст:

- Значения предварительно заполняются, когда это возможно, то есть для элементов данных, требующих агента, информация берется из выбранного интерфейса агента узла сети.
- Для элементов данных шаблона значения нужно вводить вручную.
- Значения макросов в виде обычного текста разрешаются.
- Поля, где значение (или часть значения) является секретом или макросом Vault, остаются пустыми и должны быть введены вручную. Если какой-либо параметр элемента данных содержит значение секретного макроса, отображается следующее предупреждение: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- Поля отключаются, если они не нужны в контексте типа элемента данных (например, поле адреса узла сети и поле прокси отключены для вычисляемых элементов данных)

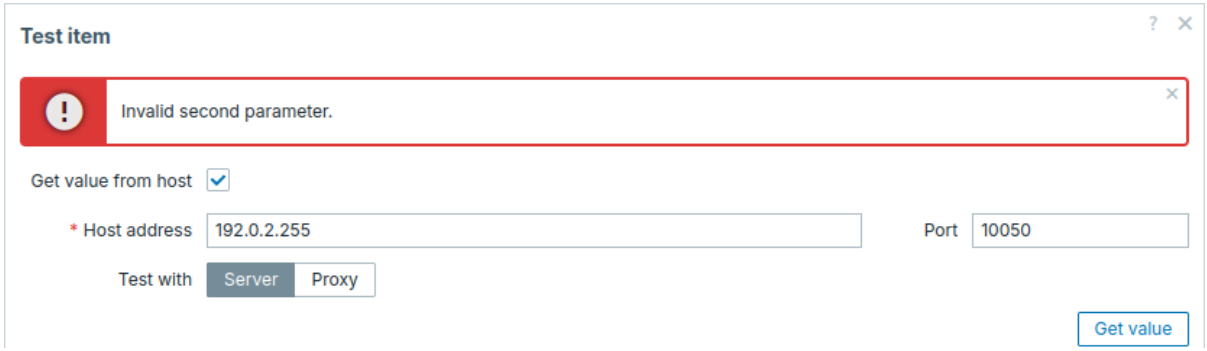
Чтобы протестировать элемент данных, нажмите *Get value*. Если значение успешно получено, оно заполнит поле *Value*, переместив текущее значение (если оно есть) в поле *Previous value*, а также вычислит поле *Prev. time*, то есть разницу во времени между двумя значениями (нажатиями), и попытается определить последовательность EOL и переключиться на CRLF, если в полученном значении обнаружено "\n\r".

Нажмите *Get value and test*, чтобы протестировать предварительную обработку.



Значения, полученные с узла сети, и результаты тестирования обрезаются до максимального размера 512KB при отправке на веб-интерфейс. Если результат тестирования обрезан, отображается значок предупреждения; наведите на него указатель, чтобы увидеть подробности. Если значение обрезано, при последующих тестах, сравнивающих большие значения (например, JSON), может не пройти проверка типа данных; в таких случаях отображается сообщение об ошибке. Обратите внимание, что данные объемом более 512KB по-прежнему полностью обрабатываются сервером Zabbix.

Если конфигурация неверна, отображается сообщение об ошибке с описанием возможной причины.



Успешно полученное значение с узла сети также можно использовать для тестирования шагов предварительной обработки.

Кнопки формы

Кнопки в нижней части формы позволяют выполнять несколько операций.

Add	Добавить элемент данных. Эта кнопка доступна только для новых элементов данных.
Update	Обновить свойства элемента данных.
Clone	Создать другой элемент данных на основе свойств текущего элемента данных.
Execute now	Немедленно выполнить проверку нового значения элемента данных. Поддерживается только для **пассивных** проверок (см. подробнее). *Примечание*: при немедленной проверке значения кэш конфигурации не обновляется, поэтому значение не будет отражать самые последние изменения конфигурации элемента данных.
Test	Проверить корректность конфигурации элемента данных, получив значение.
Clear history and trends	Удалить историю и тренды элемента данных.
Delete	Удалить элемент данных.
Cancel	Отменить редактирование свойств элемента данных.

Ограничения на данные элемента данных

Ограничения на данные элемента данных зависят от типа данных элемента данных и используемой СУБД.

Числовые значения (без знака) сохраняются без изменений. Если получено значение с плавающей точкой, сохраняется только целая часть (например, 1.23 → 1).

Числовые значения (float) сохраняются без изменений и поддерживают точность примерно от 15 до 17 цифр, с диапазоном примерно от -1.79E+308 до 1.79E+308. Также поддерживаются значения в научной нотации (например, 1.23E+7, 1e308, 1.1E-4).

Текстовые значения усекаются перед сохранением, чтобы соответствовать ограничению типа значения в базе данных:

База данных	Тип информации		
	Character	**Log**	**Text**
MySQL	255 символов	65536 байт	65536 байт
PostgreSQL	255 символов	65536 символов	65536 символов
SQLite (только Zabbix proxy)	255 символов	65536 символов	65536 символов

Двоичные значения ограничены 16 МиБ (16777216 байт). Значения, превышающие этот предел, отбрасываются, и соответствующая ошибка отображается в веб-интерфейсе.

Значения с типом данных JSON ограничены 128 МиБ (134217728 байт). Значения, превышающие этот предел, отбрасываются, и соответствующая ошибка отображается в веб-интерфейсе. Для MySQL/MariaDB рекомендуется настроить ежедневные разделы для таблицы `history_json` (используется для хранения значений JSON), так как она может быстро увеличиваться и надолго блокировать housekeeper. Если в вашем сценарии используются большие значения JSON (1 МиБ или больше), см. как настроить систему для поддержки больших значений JSON.

Когда Zabbix записывает любые данные элемента данных в базу данных, он использует запросы INSERT и ведет их журнал (если для `DebugLevel` установлено значение 4 или 5). Большие записи журнала усекаются до 64 КБ.

Ограничение пользовательского скрипта

Доступная длина пользовательского скрипта зависит от используемой базы данных:

Database	Limit in characters	Limit in bytes
MySQL	65535	65535
PostgreSQL	65535	not limited
SQLite (only Zabbix proxy)	65535	not limited

Тайм-аут элемента данных

Тайм-аут элемента данных определяет, как долго Zabbix должен ждать, прежде чем прервать проверку как неудачную.

Когда достигается тайм-аут, проверка прерывается, даже если получение данных еще не завершено. Если данные получены частично, элемент данных становится неподдерживаемым, и в журнале записывается сообщение об ошибке (например, если данные успешно собраны только для одного из нескольких OID в SNMP-проверке).

Для многих типов элементов данных можно задать **гибкие** тайм-ауты элементов данных:

- для одного элемента данных
- для типа элемента данных (на уровне прокси)
- для типа элемента данных (на глобальном уровне)

Пользовательский тайм-аут *для одного элемента данных* полезен, если нужно указать более длительный тайм-аут для конкретного элемента данных, сохранив при этом низкие тайм-ауты для остальных элементов данных.

Приоритет тайм-аутов

1. Тайм-аут одного элемента данных имеет приоритет над любыми другими тайм-аутами.
2. Тайм-ауты на уровне прокси имеют приоритет над глобальными тайм-аутами.

Тайм-аут из конфигурации Zabbix сервера или прокси **не** влияет на проверки, для которых заданы гибкие тайм-ауты.

Поддержка гибких тайм-аутов

Гибкие тайм-ауты элементов данных поддерживаются для следующих типов элементов данных:

- агент Zabbix (как пассивные, так и активные проверки)
- SNMP agent (кроме устаревших SNMP-проверок[1])
- Simple check (кроме `icmpping*` и элементов данных VMware[2])
- SSH agent
- Telnet agent
- External check
- Database monitor

- Script
- HTTP agent
- Browser

[1] Для устаревших SNMP-проверок (опрос одного OID) применяются настройки тайм-аута из конфигурации сервер или прокси.
 [2] Для элементов данных `icmpping*` значение тайм-аута указывается непосредственно в ключе элемента данных; также действует жестко заданный максимальный возможный тайм-аут 600 секунд. Элементы мониторинга VMware используют собственный параметр VMwareTimeout из конфигурации сервер или прокси.

Неподдерживаемые элементы данных

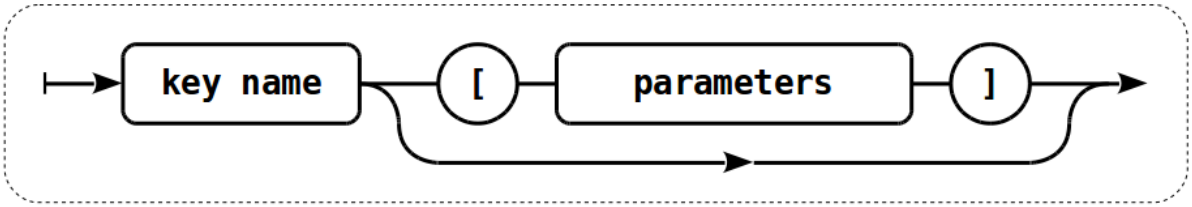
Неподдерживаемые элементы данных отображаются со статусом *Не поддерживается* и по-прежнему повторно проверяются с их стандартным *Интервалом обновления*.

Элемент данных становится неподдерживаемым, если его значение по какой-либо причине не удается получить (например, ошибка соединения, не настроены pollers для обработки элемента данных). Кроме того, элементы данных, которые не получают исторические данные, остаются в статусе *Не поддерживается*. Это ожидаемое поведение — элемент данных переходит в статус *Нормальный* только тогда, когда получает новые корректные исторические данные.

Как и проблемы, неподдерживаемые элементы данных переоцениваются только при получении новых данных — даже если для этого элемента данных больше нет исторических данных. Иными словами, элементы данных и триггеры изменяют свое состояние исключительно при получении новых данных. Однако если выражение триггера включает функцию date and time и/или nodata(), триггер будет пересчитываться каждые 30 секунд (подробности см. в разделе время вычисления триггеров).

Формат ключа элемента данных

Формат ключа элемента данных, включая параметры ключа, должен соответствовать правилам синтаксиса. Следующие иллюстрации показывают поддерживаемый синтаксис. Допустимые элементы и символы в каждой точке можно определить, следуя по стрелкам — если до некоторого блока можно добраться по линии, он разрешен, если нет — он не разрешен.



Чтобы составить корректный ключ элемента данных, сначала указывают имя ключа, затем можно выбрать, будут ли параметры или нет — как показано двумя линиями, по которым можно следовать.

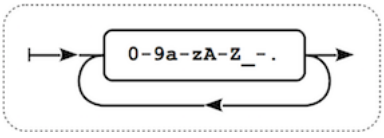
Имя ключа

Имя ключа имеет ограниченный диапазон разрешённых символов, которые просто следуют друг за другом. Разрешённые символы:

`0-9a-zA-Z_-.`

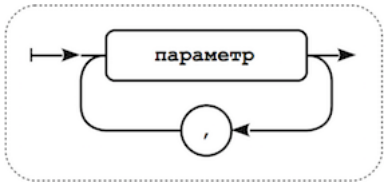
Что означает:

- все цифры;
- все латинские буквы в нижнем регистре;
- все латинские буквы в верхнем регистре;
- подчёркивание;
- минус;
- точка.



Параметры ключа

Ключ элемента данных может принимать несколько параметров, которые должны быть разделены запятой.



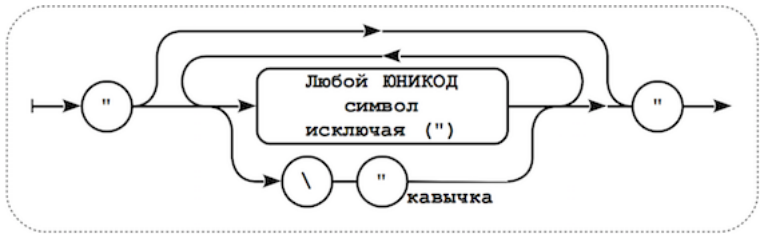
Каждый параметр ключа может быть одним из: заключённая в кавычки строка, строка без кавычек, массив.



Параметр можно оставить пустым, и тогда будет использоваться значение по умолчанию. В этом случае Вы должны добавить соответствующее количество запятых, если какой-то из дальнейших параметров все же нужно задать. Например, ключ элемента данных **icmpping[„200„500]** будет указывать на то, что период проверки ping 200 миллисекунд, время ожидания — 500 миллисекунд, а все остальные параметры будут заполнены значениями по умолчанию.

Параметр - строка, заключённая в кавычки

Если параметр ключа — это строка, заключённая в кавычки, тогда разрешён любой символ в Юникод. Если строка параметра ключа содержит кавычки, этот параметр должен быть заключён в кавычки, а каждая кавычка, являющаяся частью строки параметра, должна экранироваться символом обратной косой черты (\). Если строка параметра ключа содержит запятую, то этот параметр должен быть заключён в кавычки.

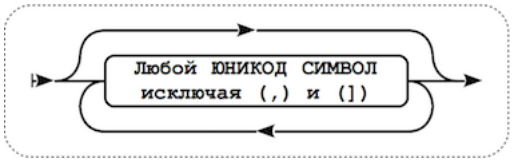


Warning:
Чтобы заключить параметры ключа элемента данных в кавычки, используйте только двойные кавычки. Одинарные кавычки не поддерживаются.

Attention:
Многоуровневые массивы параметров, например `[a,[b,[c,d]],e]`, не разрешены.

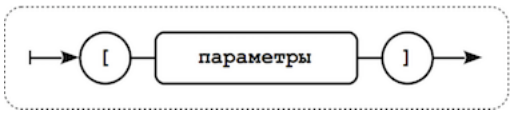
Параметр - строка, не заключённая в кавычки

Если параметр ключа — это строка без кавычек, тогда разрешён любой символ в Юникод, за исключением запятой и правой квадратной скобки (]). Параметр, который не заключён в кавычки, не может начинаться с левой квадратной скобки ([).



Параметр - массив

Если параметр ключа — это массив, тогда он должен быть заключён в квадратные скобки, в которых каждый индивидуальный параметр следует один за другим, согласно правилам и синтаксису.



Пользовательские интервалы

Обзор

Можно создавать пользовательские правила, определяющие время проверки элемента данных. Для этого предусмотрены два метода: *гибкие интервалы*, позволяющие переопределить интервал обновления по умолчанию, и *планирование*, при котором проверка элемента данных может выполняться в определенное время или в заданной последовательности.

Гибкие интервалы

Гибкие интервалы позволяют переопределить интервал обновления по умолчанию для определенных периодов времени.

Гибкий интервал задается с помощью параметров *Interval* и *Period*, где:

- *Interval* - интервал обновления для указанного периода времени. Поддерживаются суффиксы времени, например: 30s, 1m, 2h, 1d.
- *Period* - период времени, в течение которого активен гибкий интервал (подробное описание формата *Period* см. в разделе периоды времени).

Если несколько гибких интервалов перекрываются, для перекрывающегося периода используется наименьшее значение *Interval*. Обратите внимание: если наименьшее значение среди перекрывающихся гибких интервалов равно «0», опрос выполняться не будет. За пределами гибких интервалов используется интервал обновления по умолчанию.

Обратите внимание: если гибкий интервал равен продолжительности периода, элемент данных будет проверен ровно один раз. Если гибкий интервал больше периода, элемент данных может быть проверен один раз или не проверен вообще (поэтому такая конфигурация не рекомендуется). Если гибкий интервал меньше периода, элемент данных будет проверен как минимум один раз.

Если для гибкого интервала задано значение «0», опрос элемента данных в течение этого периода выполняться не будет. После его завершения опрос возобновится в соответствии с параметром *Update interval* по умолчанию. Примеры:

Interval	Period	Description
10	1-5,09:00-18:00	Элемент данных будет проверяться каждые 10 секунд в рабочие часы.
0	1-7,00:00-7:00	Элемент данных не будет проверяться ночью.
0	7-7,00:00-24:00	Элемент данных не будет проверяться по воскресеньям.
60	1-7,12:00-12:01	Элемент данных будет проверяться ежедневно в 12:00. Обратите внимание: ранее это использовалось как обходное решение для запланированных проверок; для таких проверок рекомендуется использовать интервалы планирования.

Интервалы по расписанию

Интервалы по расписанию используются для опроса элементов данных в заданное время. Тогда как переменные интервалы разрабатывались для переопределения интервала обновления элементов данных по умолчанию, интервалы по расписанию используются для обозначения независимого расписания опроса, которое выполняется параллельно.

Интервал по расписанию задается как: `md< >wd< >h< >m< >s< >` где:

- **md** — дни месяца
- **wd** — дни недели
- **h** — часы
- **m** — минуты
- **s** — секунды

`< >` используется для того, чтобы указать значения по своему префиксу (дни, часы, минуты, секунды) и задается как: `[< >[-< >]][/< >][,< >]` где:

- `< >` и `< >` определяют диапазон соответствующих значений (включительно). Если `< >` не указывается, тогда фильтр совпадает с диапазоном `< >` — `< >`. Если `< >` также не указан, тогда фильтр совпадает со всеми возможными значениями.
- `< >` определяет пропуски количества значений через диапазон. По умолчанию значение `< >` равно 1, что означает, что все значения указанного диапазона совпадают.

Хотя добавление фильтра опционально, по крайней мере один фильтр должен использоваться. Фильтр должен определять либо диапазон, либо значение *<шага>*.

Пустой фильтр соответствует либо «0», если фильтр более низкого уровня не определён, либо, в противном случае, всем возможным значениям. Например, если фильтр часа опущен, то будет соответствовать только «0» в случае, если фильтры минут и секунд также опущены; в противном случае пустой фильтр часа будет соответствовать всем значениям часа.

Допустимые значения < > и < > по их соответствующему префиксу фильтра:

Префикс	Описание	*<от>*	*<до>*
md	Дни месяца	1—31	1—31
wd	Дни недели	1—7	1—7
h	Часы	0—23	0—23
m	Минуты	0—59	0—59
s	Секунды	0—59	0—59

Значение < > должно быть меньше или равно значению < >. Значение < > должно быть больше или равно 1 и меньше или равно < > - < >.

К однозначным цифрам значений дней месяца, часов, минут и секунд можно добавлять префикс 0. Например, `md01-31` и `h/02` являются допустимыми интервалами, но `md01-031` и `wd01-07` — нет.

В веб-интерфейсе Zabbix несколько интервалов по расписанию вводятся в отдельных строках. В Zabbix API они складываются в одну строку с разделителем в виде точки с запятой (`;`).

Если время совпадает по нескольким интервалам, опрос выполнится только раз. Например, `wd1h9;h9` в понедельник в 9 утра выполнится только один раз.

Примеры:

Интервал	Когда выполняется
m0-59	каждую минуту
h9-17/2	каждые 2 часа, начиная с 9:00 (9:00, 11:00 ...)
m0,30 или m/30	каждый час в чч:00 и чч:30
m0,5,10,15,20,25,30,35,40,45,50,55 или m/5	каждые пять минут
wd1-5h9	каждый день с понедельника по Пятницу в 9:00
wd1-5h9-18	каждый день с понедельник по Пятницу в 9:00,10:00,...,18:00
h9,10,11 или h9-11	каждый день в 9:00, 10:00 и 11:00
md1h9m30	каждое первое число каждого месяца в 9:30
md1wd1h9m30	каждое первое число каждого месяца в 9:30, если этот день понедельник
h9m/30	каждый день в 9:00, 9:30
h9m0-59/30	каждый день в 9:00, 9:30
h9,10m/30	каждый день в 9:00, 9:30, 10:00, 10:30
h9-10m30	каждый день в 9:30, 10:30
h9m10-40/30	каждый день в 9:10, 9:40
h9,10m10-40/30	каждый день в 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	каждый день в 9:10, 9:40, 10:10, 10:40
h9m10-40	каждый день в 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	каждый день в 9:10, 9:11, 9:12, ... 9:40
h9-12,15	каждый день в 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	каждый день в 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	каждый день в 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	каждый день в 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*особый синтаксис API*)	каждый день в 9:00, 9:30, 10:00
h9m/30 h10 (*добавьте это отдельной строкой в веб-интерфейсе*)	каждый день в 9:00, 9:30, 10:00

Выравнивание часовых поясов для прокси и агента

Обратите внимание, что Zabbix прокси и агент, когда обрабатывают интервалы по расписанию, используют свои местные часовые пояса.

По этой причине, когда интервалы по расписанию применяются к элементам данных, которые отслеживаются через прокси, либо к активным элементам данных агента, то рекомендуется настроить часовой пояс соответствующих прокси

либо агента таким же, как и на сервере Zabbix, в противном случае очередь может неправильно сообщать о задержках элементов данных.

Часовой пояс для Zabbix прокси или агента можно настроить при помощи переменной окружения TZ в файле юнита systemd:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

Предобработка значений элемента данных

Обзор

Предварительная обработка позволяет применять преобразования к полученным значениям элементов данных перед сохранением их в базе данных. Эти преобразования/шаги предобработки выполняются Zabbix сервером или прокси (если элементы данных отслеживаются через прокси).

Этот функционал поддерживает множество вариантов использования, таких как:

- преобразование байтов в биты (например, умножая значения сетевого трафика на «8»);
- расчёт посекундной статистики постепенно возрастающих значений;
- применение регулярных выражений для извлечения или изменения значений;
- выполнение пользовательских скриптов для обработки значений;
- отбрасывание неизменившихся значений для оптимизации хранения базы данных.

Для элемента данных можно настроить один или несколько шагов предварительной обработки. Эти шаги выполняются в том порядке, в котором они настроены.

Attention:
Если шаг предварительной обработки завершается неудачей, элемент данных станет неподдерживаемым. Этого можно избежать с помощью пользовательской обработки ошибок с использованием параметра *Другое при ошибке* (доступной для большинства преобразований), что позволяет вам отбрасывать значения или устанавливать пользовательские значения.

 Для элементов данных типа «журнал» метаданные журнала (без значения) всегда будут сбрасывать неподдерживаемое состояние элемента данных и снова делать элемент данных поддерживаемым. Это происходит, даже если первоначальная ошибка произошла после получения значения журнала от агента.

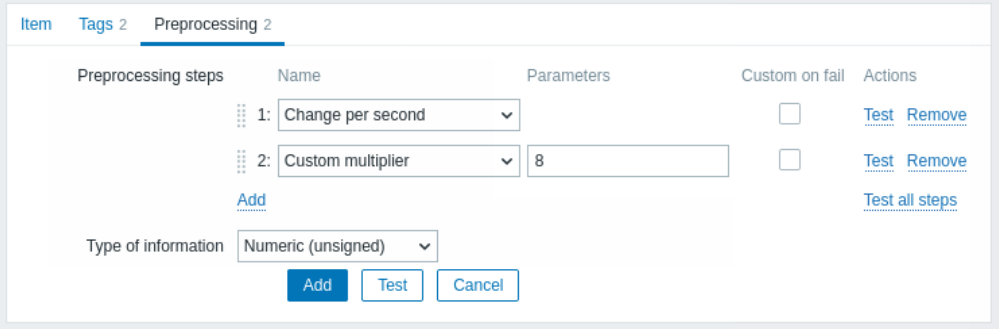
Все значения, передаваемые в предварительную обработку, изначально рассматриваются как строки. Преобразование в требуемый тип значения (как определено в конфигурации элемента данных) выполняется в конце конвейера предобработки. Однако, определённые шаги предварительной обработки могут инициировать более ранние преобразования, если это необходимо. Подробную техническую информацию смотрите в разделе Подробности предварительной обработки.

Чтобы убедиться, что ваша конфигурация предварительной обработки работает так, как ожидается, можно её протестировать.

Смотрите также: Примеры предварительной обработки

Настройка

Шаги предварительной обработки задаются на вкладке **Предварительная обработка** формы настройки элемента данных.



Нажмите *Добавить*, чтобы выбрать поддерживаемое преобразование.

Поле *Тип информации* отображается в нижней части вкладки, если задан хотя бы один шаг предварительной обработки. При необходимости тип информации можно изменить, не покидая вкладку *Предварительная обработка*. Подробное описание параметров см. в разделе Создание элемента данных.

Поддерживаемые преобразования

Ниже перечислены все поддерживаемые преобразования.
Нажмите на название преобразования, чтобы просмотреть подробную информацию о нем.

Название	Описание	Тип
Регулярное выражение	Сопоставить значение с регулярным выражением и заменить его требуемым результатом.	Текст
Замена	Найти строку поиска и заменить ее другой строкой (или ничем).	
Обрезка	Удалить указанные символы из начала и конца значения.	
Обрезка справа	Удалить указанные символы из конца значения.	
Обрезка слева	Удалить указанные символы из начала значения.	
XML XPath	Извлечь значение или фрагмент из данных XML с помощью функциональности XPath.	Структурированные данные
JSON Path	Извлечь значение или фрагмент из данных JSON с помощью функциональности JSONPath.	
CSV в JSON	Преобразовать данные CSV-файла в формат JSON.	
XML в JSON	Преобразовать данные в формате XML в JSON.	
Значение SNMP walk	Извлечь значение по указанному имени OID/MIB и применить параметры форматирования.	SNMP
SNMP walk в JSON	Преобразовать значения SNMP в JSON.	
Получение значения SNMP	Применить параметры форматирования к полученному значению SNMP.	
Пользовательский множитель	Умножить значение на указанное целое число или число с плавающей точкой.	Арифметика
Простое изменение	Вычислить разницу между текущим и предыдущим значением.	Изменение
Изменение в секунду	Вычислить скорость изменения значения (разницу между текущим и предыдущим значением) в секунду.	
Логическое значение в десятичное	Преобразовать значение из логического формата в десятичный.	Системы счисления
Восьмеричное в десятичное	Преобразовать значение из восьмеричного формата в десятичный.	
Шестнадцатеричное в десятичное	Преобразовать значение из шестнадцатеричного формата в десятичный.	
JavaScript	Ввести код JavaScript.	Пользовательские скрипты
В диапазоне	Определить диапазон, в котором должно находиться значение.	Проверка
Соответствует регулярному выражению	Указать регулярное выражение, которому должно соответствовать значение.	
Не соответствует регулярному выражению	Указать регулярное выражение, которому не должно соответствовать значение.	
Проверка ошибки в JSON	Проверить наличие сообщения об ошибке на уровне приложения, расположенного по JSONPath.	
Проверка ошибки в XML	Проверить наличие сообщения об ошибке на уровне приложения, расположенного по XPath.	
Проверка ошибки с помощью регулярного выражения	Проверить наличие сообщения об ошибке на уровне приложения с помощью регулярного выражения.	
Проверка неподдерживаемого значения	Проверить, не удалось ли получить значение элемента данных.	
Отбрасывать неизменившиеся	Отбрасывать значение, если оно не изменилось.	Регулирование
Отбрасывать неизменившиеся с контролем доступности	Отбрасывать значение, если оно не изменилось в течение заданного периода времени.	

Название	Описание	Тип
Шаблон Prometheus	Использовать следующий запрос для извлечения требуемых данных из метрик Prometheus.	Prometheus
Prometheus в JSON	Преобразовать требуемые метрики Prometheus в JSON.	

Если для выбранного преобразования включен флажок *Пользовательское действие при ошибке*, его можно отметить и указать пользовательские параметры обработки ошибок: *Отбросить значение* (отбросить значение), *Установить значение* (установить указанное значение) или *Установить ошибку* (установить указанное сообщение об ошибке).
Если шаг предварительной обработки завершается ошибкой, элемент данных не станет неподдерживаемым, если выбран параметр *Отбросить значение* или *Установить значение*.

Обратите внимание, что для шагов предварительной обработки *Изменение* и *Регулирование* Zabbix должен запоминать последнее значение, чтобы при необходимости вычислять или сравнивать новое значение.
Эти предыдущие значения обрабатываются менеджером предварительной обработки.
Если сервер Zabbix или прокси перезапущены либо в шаги предварительной обработки внесены изменения, последнее значение соответствующего элемента данных сбрасывается, что приводит к следующему:

- для шагов *Простое изменение* и *Изменение в секунду* — следующее значение будет проигнорировано, поскольку отсутствует предыдущее значение, относительно которого можно вычислить изменение;
- для шагов *Отбрасывать неизменившиеся* и *Отбрасывать неизменившиеся с контролем доступности* — следующее значение никогда не будет отброшено, даже если согласно правилам отбрасывания оно должно быть отброшено.

Регулярное выражение

Сопоставляет значение с регулярным выражением и заменяет его требуемым результатом.

Параметры:

- **pattern** - регулярное выражение;
- **output** - шаблон форматирования результата. Экранирующая последовательность \N (где N=1...9) заменяется на N-ю найденную группу. Экранирующая последовательность \0 заменяется на найденный текст.

Примечания:

- Если входное значение не соответствует регулярному выражению, элемент данных станет неподдерживаемым;
- Регулярное выражение поддерживает извлечение не более 10 захваченных групп с помощью последовательности \N;
- Некоторые существующие примеры приведены в разделе регулярные выражения.

Замена

Найдите строку поиска и замените ее другой строкой (или ничем).

Параметры:

- **строка поиска** — строка, которую необходимо найти и заменить; учитывается регистр (обязательный параметр);
- **замена** — строка, которой необходимо заменить строку поиска. Строка замены также может быть пустой, что позволяет удалить найденную строку поиска.

Комментарии:

- Все вхождения строки поиска будут заменены;
- Для поиска или замены переводов строк, возвратов каретки, табуляций и пробелов можно использовать escape-последовательности ”\n \r \t \s”; обратную косую черту можно экранировать как ”\\”; escape-последовательности можно экранировать как ”\\n”;
- Экранирование переводов строк, возвратов каретки и табуляций автоматически выполняется во время низкоуровневого обнаружения.

Обрезать (Trim)

Удалить указанные символы с начала и конца значения.

Обрезать справа (Right trim)

Удалить указанные символы с конца значения.

Обрезать слева (Left trim)

Удалить указанные символы с начала значения.

XML XPath

Извлечение значения или фрагмента из XML-данных с помощью функциональности XPath.

Комментарии:

- Для работы этого параметра сервер Zabbix (или прокси Zabbix) должен быть скомпилирован с поддержкой libxml;
- Пространства имён не поддерживаются.

Примеры:

```
number(/document/item/value) #     «10»   <document><item><value>10</value></item></document>
number(/document/item/@attribute) #     «10»   <document><item attribute="10"></item></document>
/document/item #     «<item><value>10</value></item>»   <document><item><value>10</value></item></document
```

JSON Path

Извлечение значения или фрагмента из данных JSON с использованием функциональности JSONPath.

CSV в JSON

Сконвертировать данные файла CSV в формат JSON.

Для получения дополнительной информации смотрите: Преобразование CSV в JSON.

XML в JSON

Преобразование данных из формата XML в JSON.

Дополнительную информацию см. в разделе: Правила сериализации.

Значение SNMP walk

Извлечение значения по указанному имени OID/MIB и применение параметров форматирования:

- **Без изменений** - возвращать hex-строку как экранированную hex-строку (*обратите внимание*, что подсказки отображения по-прежнему применяются);
- **UTF-8 из hex-STRING** - преобразовать hex-строку в строку UTF-8;
- **MAC из hex-STRING** - проверить hex-строку как MAC-адрес и вернуть корректную строку MAC-адреса (где `' '` заменяются на `':'`);
- **Целое число из BITS** - преобразовать первые 8 байт битовой строки, представленной в виде последовательности шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное беззнаковое целое число. В битовых строках длиной более 8 байт последующие байты игнорируются.

Обход SNMP в JSON

Преобразование значений SNMP в JSON.

Укажите имя поля в JSON и соответствующий путь к OID SNMP. Значения полей будут заполнены значениями из указанного пути к OID SNMP.

Комментарии:

- Доступны параметры форматирования значений, аналогичные шагу *Значение обхода SNMP*;
- Этот шаг предварительной обработки можно использовать для обнаружения OID SNMP.

Получение значения SNMP

Применить параметры форматирования к значению, полученному по SNMP:

- **UTF-8 из Hex-STRING** — преобразовать шестнадцатеричную строку в строку UTF-8;
- **MAC из Hex-STRING** — проверить шестнадцатеричную строку как MAC-адрес и вернуть корректную строку MAC-адреса (при этом `' '` заменяются на `':'`);
- **Целое число из BITS** — преобразовать первые 8 байт битовой строки, представленной в виде последовательности шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-разрядное беззнаковое целое число. В битовых строках длиной более 8 байт последующие байты игнорируются.

Пользовательский множитель

Умножает значение на указанное целое число или число с плавающей точкой.

Комментарии:

- Используйте этот параметр для преобразования значений, полученных в КБ, МБ/с и т. д., в Б, Б/с. В противном случае Zabbix не сможет правильно установить префиксы (K, M, Г и т. д.).
- Обратите внимание: если тип информации элемента данных — *Числовой (без знака)*, входящие значения с дробной частью будут усечены (то есть «0.9» превратится в «0») до применения пользовательского множителя;

- Если для элементов данных с типом информации *Числовой (без знака)* используется пользовательский множитель или сохраняется значение как *Изменение в секунду*, а результирующее вычисленное значение фактически является числом с плавающей точкой, вычисленное значение все равно принимается как корректное: десятичная часть отбрасывается, а значение сохраняется как целое число;
- Поддерживаются: экспоненциальная запись, например 1e+70; пользовательские макросы и макросы LLD; строки, содержащие макросы, например `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Макросы должны разрешаться в целое число или число с плавающей точкой.

Простое изменение

Вычисляет разницу между текущим и предыдущим значением.

Комментарии:

- Этот шаг может быть полезен для измерения постоянно увеличивающегося значения;
- Вычисляется как **value**-**prev_value**, где *value* — текущее значение; *prev_value* — ранее полученное значение;
- Для одного элемента данных допускается только одна операция изменения («Простое изменение» или «Изменение в секунду»);
- Если текущее значение меньше предыдущего, Zabbix отбрасывает эту разницу (ничего не сохраняет) и ожидает следующее значение.

Изменение в секунду

Рассчитать скорость изменения значения (разницу между текущим и предыдущим значением) в секунду.

Комментарии:

- Этот шаг полезен для расчета скорости в секунду для постоянно увеличивающегося значения;
- Поскольку в результате этого расчета могут получаться числа с плавающей запятой, рекомендуется установить для параметра «Тип информации» значение *Числовой (с плавающей запятой)*, даже если исходные входящие значения являются целыми числами. Это особенно важно для небольших чисел, где имеет значение десятичная часть. Если значения с плавающей запятой являются большими и могут превысить длину поля `float`, в результате чего может быть потеряно все значение, рекомендуется использовать *Числовой (без знака)* и таким образом отбрасывать только десятичную часть;
- Вычисляется как (**value**-**prev_value**)/(**time**-**prev_time**), где *value* — текущее значение; *prev_value* — ранее полученное значение; *time* — текущая временная метка; *prev_time* — временная метка предыдущего значения;
- Для одного элемента данных разрешена только одна операция изменения («Простое изменение» или «Изменение в секунду»);
- Если текущее значение меньше предыдущего, Zabbix отбрасывает эту разницу (ничего не сохраняет) и ожидает следующее значение. Это помогает корректно работать, например, с переполнением 32-разрядных счетчиков SNMP.

Логическое значение в десятичное

Преобразование значения из логического формата в десятичный.

Комментарии:

- Текстовое представление преобразуется в 0 или 1. Таким образом, «TRUE» сохраняется как 1, а «FALSE» — как 0. Все значения сопоставляются без учета регистра. В настоящее время распознаются следующие значения: для *TRUE* — true, t, yes, y, on, up, running, enabled, available, ok, master; для *FALSE* — false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Кроме того, любое ненулевое числовое значение считается значением TRUE, а нулевое — значением FALSE.

Из восьмеричной системы в десятичную

Преобразует значение из восьмеричного формата в десятичный.

Преобразование шестнадцатеричного числа в десятичное

Преобразует значение из шестнадцатеричного формата в десятичный.

JavaScript

Введите код JavaScript в модальном редакторе, который открывается при нажатии на поле параметра или на значок карандаша рядом с ним.

> **Attention:**
> Не используйте необъявленные присваивания в предварительной обработке JavaScript. Используйте `var` для объявления локальных переменных.

Комментарии:

- Доступная длина JavaScript зависит от используемой базы данных;
- Для получения дополнительной информации смотрите: Предобработка Javascript.

В диапазоне

Определите диапазон, в котором должно находиться значение, указав минимальное и максимальное значения (включительно).

Комментарии:

- Допускаются числовые значения (включая любое количество цифр, необязательную десятичную и необязательную экспоненциальную часть, отрицательные значения);
- Минимальное значение должно быть меньше максимального;
- Должно существовать хотя бы одно значение;
- Можно использовать пользовательские макросы и макросы низкоуровневого обнаружения.

Соответствует регулярному выражению

Укажите регулярное выражение, которому должно соответствовать значение.

Не соответствует регулярному выражению

Укажите регулярное выражение, которому значение не должно соответствовать.

Проверка наличия ошибки в JSON

Проверка сообщения об ошибке на уровне приложения, расположенного по JSONPath. Прекратить обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжить обработку со значением, которое было до этого шага предварительной обработки.

Комментарии:

- Эти ошибки внешней службы отображаются пользователю как есть, без добавления информации о шаге предварительной обработки;
- Ошибка не будет отображаться в случае сбоя при разборе некорректного JSON.

Проверка наличия ошибки в XML

Проверяет наличие сообщения об ошибке на уровне приложения, расположенного по адресу XPath.
Прекращает обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжает обработку со значением, которое было до этого шага предварительной обработки.

Комментарии:

- Эти ошибки внешней службы отображаются пользователю как есть, без добавления информации о шаге предварительной обработки;
- Ошибка не будет отображаться в случае сбоя при разборе некорректного XML.

Проверка ошибки с помощью регулярного выражения

Проверка сообщения об ошибке на уровне приложения с помощью регулярного выражения.
Прекратить обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжить обработку со значением, которое было до этого шага предварительной обработки.

Параметры:

- **pattern** - регулярное выражение;
- **output** - шаблон форматирования вывода.
  Экранирующая последовательность \N (где N=1…9) заменяется на N-ю найденную группу.
  Экранирующая последовательность \0 заменяется на найденный текст.

Комментарии:

- Эти ошибки внешней службы сообщаются пользователю как есть, без добавления информации о шаге предварительной обработки.

Проверка неподдерживаемого значения

Проверка того, удалось ли получить значение элемента данных. Укажите, как следует обрабатывать ошибку, на основе анализа возвращенного сообщения об ошибке.

Параметры:

- **scope** - выберите область обработки ошибки:
any error - любая ошибка;
error matches - только ошибка, соответствующая регулярному выражению, указанному в поле *pattern*;
error does not match - только ошибка, не соответствующая регулярному выражению, указанному в поле *pattern*

- **pattern** - регулярное выражение для сопоставления с ошибкой. Если в параметре scope выбрано значение *any error*, это поле не отображается. Если поле отображается, оно является обязательным.

Комментарии:

- Обычно отсутствие значения или невозможность его получить приводит к тому, что элемент данных становится неподдерживаемым. Этот шаг предварительной обработки позволяет изменить такое поведение.
- Этот шаг предварительной обработки проверяет только невозможность получить значение элемента данных. Например, он не проверяет, соответствует ли тип полученного значения (например, строка) типу информации элемента данных (например, числовой); подробную информацию см. в разделе Примеры предварительной обработки. При несовпадении типов элемент данных все еще может стать неподдерживаемым после выполнения всех шагов предварительной обработки. Для проверки несовпадения типов можно, например, использовать шаг предварительной обработки *Custom multiplier*; см. Примеры предварительной обработки.
- Захват групп регулярного выражения поддерживается в полях *Set value to* или *Set error to*. Используйте \N (где N=1…9), чтобы получить N-ю совпавшую группу; используйте \0, чтобы получить совпавший текст;
- Эти шаги всегда выполняются первыми и после сохранения изменений элемента данных размещаются выше всех остальных шагов предварительной обработки;
- Поддерживается несколько шагов *Check for not supported value* в указанном порядке. Шаг для *any error* автоматически размещается последним шагом в этой группе.

Отбрасывать не изменившееся (Discard unchanged)

Отбросить значение, если оно не изменилось.

Комментарии:

- Если значение отбрасывается, оно не сохраняется в базе данных и Zabbix сервер не будет знать, что значение было получено. Выражения триггеров не будут пересчитываться, в результате не будут созданы/закрыты проблемы по связанным триггерам. Функции триггеров будут работать только на основе данных, которые фактически сохраняются в базу данных. Поскольку динамика изменений строится на основе данных в базе данных, если в течение часа не сохраняются значения, то для этого часа также не будет данных динамики изменений.
- Для элемента данных можно указать только один параметр троттлинга.

Отбрасывать неизменившиеся значения с периодическим сохранением

Отбрасывать значение, если оно не изменилось в течение заданного периода времени (в секундах).

Комментарии:

- Поддерживаются положительные целые значения для указания количества секунд (минимум - 1 секунда);
- Можно использовать суффиксы времени (например, 30s, 1m, 2h, 1d);
- Можно использовать пользовательские макросы и макросы обнаружения низкого уровня;
- Если значение отбрасывается, оно не сохраняется в базе данных, и сервер Zabbix не знает, что это значение было получено. Выражения триггеров вычисляться не будут, поэтому проблемы для связанных триггеров создаваться или разрешаться не будут. Функции будут работать только на основе данных, фактически сохраненных в базе данных. Поскольку тренды строятся на основе данных в базе данных, если значение не сохранялось в течение часа, данные трендов за этот час также отсутствуют.
- Для одного элемента данных можно указать только один параметр регулирования.

Шаблон Prometheus (Prometheus pattern)

Использовать заданный запрос для извлечения необходимых данных из метрик Prometheus.

Обратитесь к проверкам Prometheus для получения более подробных сведений.

Prometheus в JSON

Преобразовать необходимые метрики Prometheus в JSON.

Обратитесь к проверкам Prometheus для получения более подробных сведений.

Поддержка макросов

Пользовательские макросы и пользовательские макросы с контекстом поддерживаются в:

- параметрах шагов предварительной обработки, включая код JavaScript;
- параметрах обработки пользовательских ошибок (поля *Задать значением* и *Задать ошибку в*).

Note:
Контекст макроса игнорируется, когда макрос заменяется своим значением. Значение макроса подставляется в код как есть, без возможности добавления дополнительного экранирования до помещения значения в код JavaScript. Обратите внимание, что в некоторых случаях это может привести к ошибкам JavaScript.

Тестирование

Смотрите тестирование предварительной обработки.

Тестирование предварительной обработки

Тестирование

Тестирование шагов предварительной обработки полезно, чтобы убедиться, что сложные конвейеры предварительной обработки дают ожидаемые результаты, не дожидаясь получения и предварительной обработки значения элемента данных.

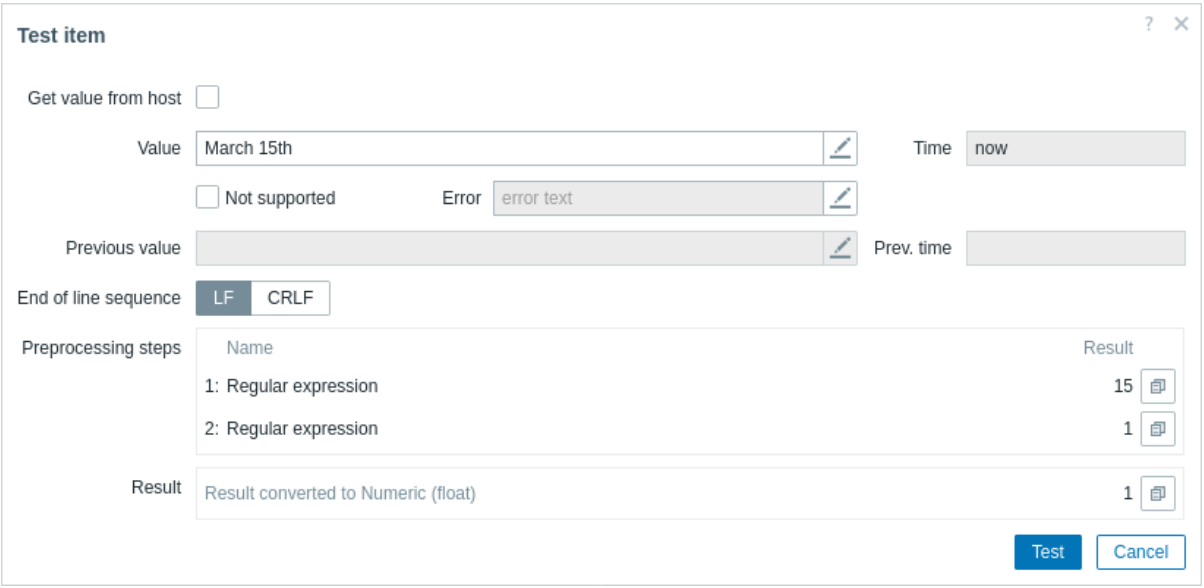


Можно выполнить тестирование:

- на основе гипотетического значения
- на основе реального значения с узла сети

Каждый шаг предварительной обработки можно тестировать отдельно, а также все шаги можно тестировать вместе. Когда вы нажимаете кнопку *Test* или *Test all steps* соответственно в блоке Actions, открывается окно тестирования.

Проверка гипотетического значения



Parameter	Description
Get value from host	Если вы хотите проверить гипотетическое значение, оставьте этот флажок снятым. См. также: Проверка реального значения.
Value	Введите значение для проверки. Щелчок в поле параметра или по кнопке просмотра/редактирования откроет окно с текстовой областью для ввода значения или блока кода.
Not supported	Установите этот флажок, чтобы проверить неподдерживаемое значение. Этот параметр полезен для проверки шага предварительной обработки *Check for not supported value*.
Error	Введите текст ошибки. Это поле доступно, когда флажок *Get value from host* снят, а *Not supported* установлен. Если флажок *Get value from host* установлен, в это поле подставляется фактическое сообщение об ошибке (только для чтения) с узла сети.
Time	Отображается время входного значения: `now` (только для чтения).
Previous value	Введите предыдущее входное значение для сравнения. Только для шагов предварительной обработки *Change* и *Throttling*.

Parameter	Description
Previous time	Введите время предыдущего входного значения для сравнения. Только для шагов предварительной обработки *Change* и *Throttling*. Значение по умолчанию основано на значении поля 'Update interval' элемента данных (если указано '1m', то это поле заполняется как `now-1m`). Если ничего не указано или у пользователя нет доступа к узлу сети, по умолчанию используется `now-30s`.
Macros	Если используются какие-либо макросы, они перечисляются вместе со своими значениями. Значения можно редактировать для целей проверки, но изменения будут сохранены только в рамках сеанса проверки.
End of line sequence	Выберите последовательность конца строки для многострочных входных значений: **LF** - последовательность LF (перевод строки) **CRLF** - последовательность CRLF (возврат каретки + перевод строки).
Preprocessing steps	Перечислены шаги предварительной обработки; результат проверки отображается для каждого шага после нажатия кнопки *Test*. Результаты проверки обрезаются до максимального размера 512KB при отправке на веб-интерфейс. Результаты проверки можно копировать (не более обрезанных 512KB). Если результат обрезан, отображается значок предупреждения. Описание предупреждения показывается при наведении курсора. Обратите внимание, что данные объемом более 512KB по-прежнему полностью обрабатываются сервером Zabbix. Если при проверке шаг завершился с ошибкой, отображается значок ошибки. Описание ошибки показывается при наведении курсора. Если для шага указан параметр "Custom on fail" и это действие выполняется, сразу после строки шага предварительной обработки появляется новая строка, показывающая, какое действие было выполнено и какой результат оно дало (ошибка или значение).
Result	Итоговый результат проверки шагов предварительной обработки отображается во всех случаях, когда все шаги проверяются вместе (при нажатии кнопки *Test all steps*). Также отображается тип преобразования к типу значения элемента данных, например `Result converted to Numeric (unsigned)`. Результаты проверки обрезаются до максимального размера 512KB при отправке на веб-интерфейс. Итоговый результат можно копировать (не более обрезанных 512KB). Если результат обрезан, отображается значок предупреждения. Описание предупреждения показывается при наведении курсора. Обратите внимание, что данные объемом более 512KB по-прежнему полностью обрабатываются сервером Zabbix.

Нажмите *Test*, чтобы увидеть результат после каждого шага предварительной обработки.

Тестовые значения сохраняются между сеансами проверки для отдельных шагов или для всех шагов, что позволяет пользователю изменять шаги предварительной обработки или конфигурацию элемента данных, а затем возвращаться в окно проверки без необходимости повторно вводить информацию. Однако при обновлении страницы значения теряются.

Проверка выполняется сервером Zabbix. Веб-интерфейс отправляет соответствующий запрос на сервер и ожидает результат. Запрос содержит входное значение и шаги предварительной обработки (с раскрытыми пользовательскими макросами). Для шагов *Change* и *Throttling* можно указать необязательные предыдущее значение и время. Сервер возвращает результаты для каждого шага предварительной обработки.

Все технические ошибки или ошибки проверки входных данных отображаются в поле ошибки в верхней части окна проверки.

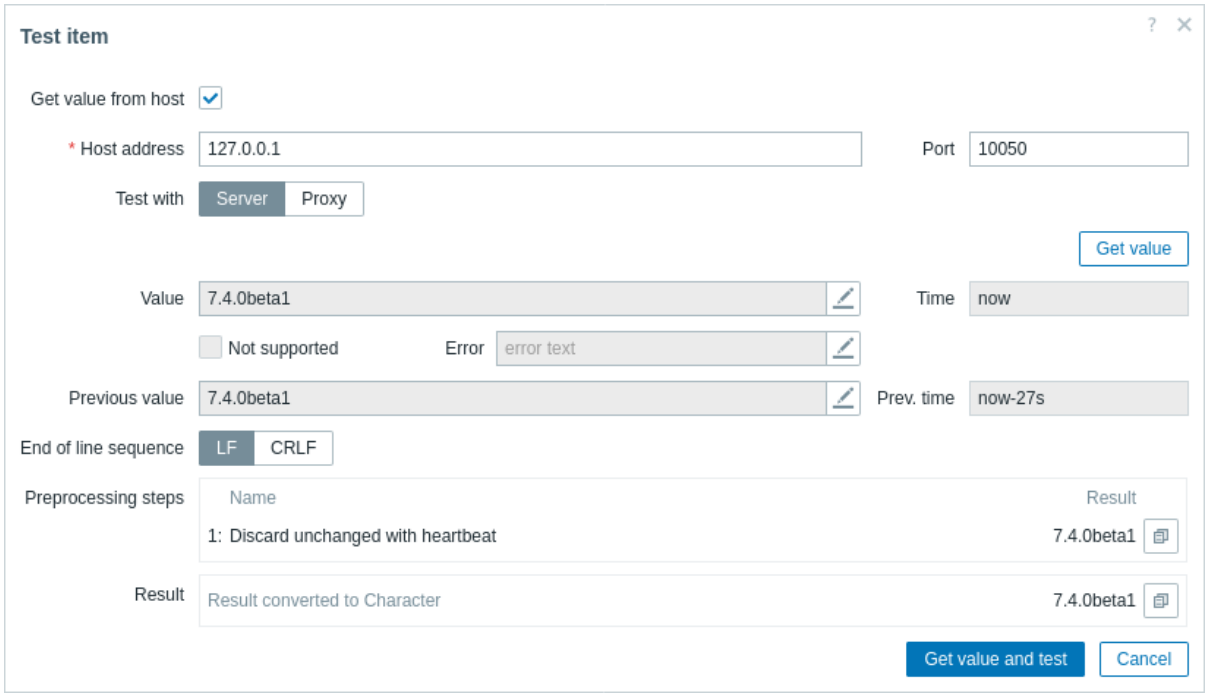
Тестирование фактического значения

Чтобы протестировать предобработку на фактическом значении:

- Установите флажок *Получить значение с узла сети*
- Введите или проверьте параметры узла сети (адрес узла сети, порт, имя прокси или отсутствие прокси) и сведения, относящиеся к элементу данных (например, community SNMPv2 или учетные данные безопасности SNMPv3). Эти поля учитывают контекст:
  - Значения предварительно заполняются, если это возможно, например для элементов данных, которым требуется агент, путем получения информации из выбранного интерфейса агента узла сети
  - Значения для элементов данных шаблона необходимо заполнять вручную
  - Значения макросов в виде обычного текста раскрываются
  - Если значение поля (или его часть) является секретом или макросом Vault, поле будет пустым, и его необходимо заполнить вручную. Если какой-либо параметр элемента данных содержит значение секретного макроса, отображается следующее предупреждение: «Элемент данных содержит пользовательские макросы с секретными значениями. Значения этих макросов необходимо ввести вручную».

- Поля отключены, если они не нужны в контексте типа элемента данных (например, поля адреса узла сети и прокси отключены для вычисляемых элементов данных)

- Нажмите *Получить значение и протестировать*, чтобы протестировать предобработку



Если в форме настройки элемента данных вы указали сопоставление значений (поле *Показать значение*), диалоговое окно тестирования элемента данных отобразит после итогового результата еще одну строку с названием *Результат с примененным сопоставлением значений*.

Параметры, относящиеся к получению фактического значения с узла сети:

Параметр	Описание
Получить значение с узла сети	Установите этот флажок, чтобы получить фактическое значение с узла сети.
Адрес узла сети	Введите адрес узла сети. Это поле автоматически заполняется адресом интерфейса узла сети, связанного с элементом данных.
Порт	Введите порт узла сети. Это поле автоматически заполняется портом интерфейса узла сети, связанного с элементом данных.
Дополнительные поля для интерфейсов SNMP (версия SNMP, community SNMP, имя контекста и т. д.)	Дополнительные сведения о настройке интерфейса SNMP (v1, v2 и v3) см. в разделе Настройка мониторинга SNMP. Эти поля автоматически заполняются из интерфейса узла сети, связанного с элементом данных.
Прокси	Укажите прокси, если мониторинг узла сети выполняется прокси. Это поле автоматически заполняется прокси узла сети (если он есть).
Значение	Значение, полученное с узла сети. При нажатии в поле параметра или на кнопку просмотра/редактирования открывается окно текстовой области со значением или блоком кода. Значения усекаются до максимального размера 512 КБ только в веб-интерфейсе. Если результат усечен, отображается значок предупреждения. Описание предупреждения отображается при наведении указателя мыши. Обратите внимание, что данные размером более 512 КБ по-прежнему полностью обрабатываются сервером Zabbix.

Описание остальных параметров см. выше в разделе Тестирование гипотетического значения.

Подробности предварительной обработки

Обзор

В этом разделе приведены сведения о предварительной обработке значений элементов данных.

Предварительная обработка значений элементов данных позволяет определить и выполнить правила преобразования для полученных значений элементов данных.

Предварительной обработкой управляет процесс диспетчера предварительной обработки вместе с рабочими процессами предварительной обработки, которые выполняют этапы предварительной обработки.

Все значения с предварительной обработкой, полученные от разных сборщиков данных, проходят через диспетчер предварительной обработки, прежде чем попасть в кэш истории. Для взаимодействия между сборщиками данных (опросчиками, трапперами и т. д.) и процессом предварительной обработки используется сокетное взаимодействие между процессами (IPC). Этапы предварительной обработки выполняет сервер Zabbix или прокси Zabbix (для элементов данных, которые отслеживаются прокси).

Обработка значений элементов данных

Чтобы визуализировать поток данных от источника данных до базы данных Zabbix, можно использовать следующую упрощенную диаграмму:

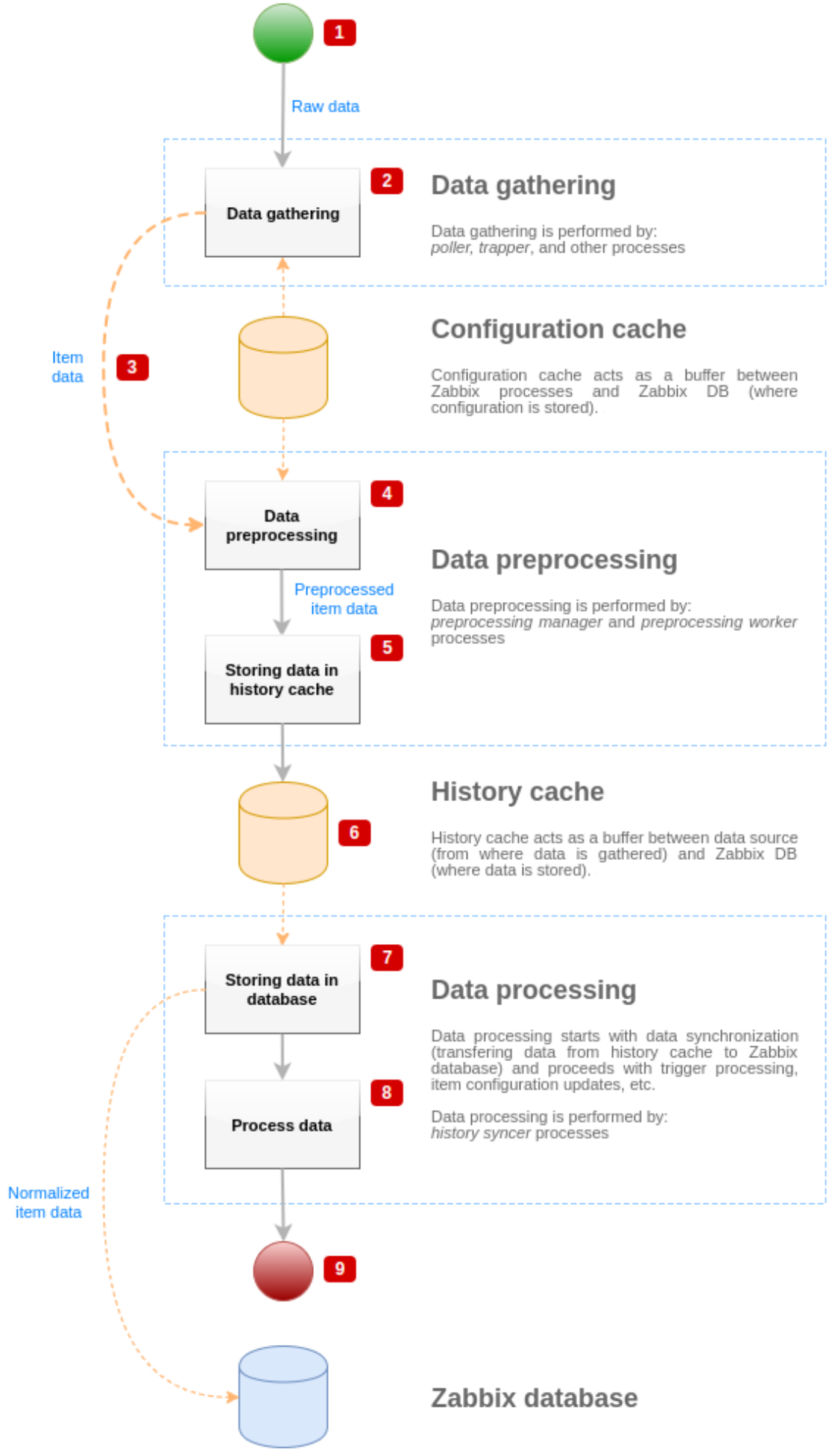

1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

Диаграмма выше показывает только процессы, объекты и действия, связанные с обработкой значений элементов данных, в **упрощенном** виде. На диаграмме не показаны изменения направления при выполнении условий, обработка ошибок или циклы. Локальный кэш данных менеджера предварительной обработки также не показан, поскольку он напрямую не влияет на поток данных. Цель этой диаграммы — показать процессы, участвующие в обработке значений элементов данных, и то, как они взаимодействуют.

- Сбор данных начинается с необработанных данных из источника данных. На этом этапе данные содержат только ID, метку времени и значение (возможны также несколько значений).
- Независимо от того, какой тип сборщика данных используется, принцип остается одинаковым для активных и пассивных проверок, для элементов данных типа trapper и т. д., поскольку меняется только формат данных и инициатор связи (либо сборщик данных ожидает соединения и данные, либо сборщик данных инициирует связь и запрашивает данные). Необработанные данные проверяются, конфигурация элемента данных извлекается из кэша конфигурации (данные дополняются конфигурационной информацией).
- Для передачи данных от сборщиков данных к менеджеру предварительной обработки используется IPC-механизм на основе сокетов. На этом этапе сборщик данных продолжает собирать данные, не ожидая ответа от менеджера предварительной обработки.
- Выполняется предварительная обработка данных. Она включает выполнение шагов предварительной обработки и обработку зависимых элементов данных.

Note:
Элемент данных может изменить свое состояние на NOT SUPPORTED во время выполнения предварительной обработки, если любой из шагов предварительной обработки завершится с ошибкой.

- Исторические данные из локального кэша данных менеджера предварительной обработки переносятся в кэш истории.
- На этом этапе поток данных останавливается до следующей синхронизации кэша истории (когда процесс синхронизации истории выполняет синхронизацию данных).
- Процесс синхронизации начинается с нормализации данных перед сохранением в базе данных Zabbix. Нормализация данных выполняет преобразование к требуемому типу элемента данных (тип, определенный в конфигурации элемента данных), включая усечение текстовых данных на основе предопределенных размеров, разрешенных для этих типов (HISTORY_STR_VALUE_LEN для string, HISTORY_TEXT_VALUE_LEN для text и HISTORY_LOG_VALUE_LEN для log values). После завершения нормализации данные отправляются в базу данных Zabbix.

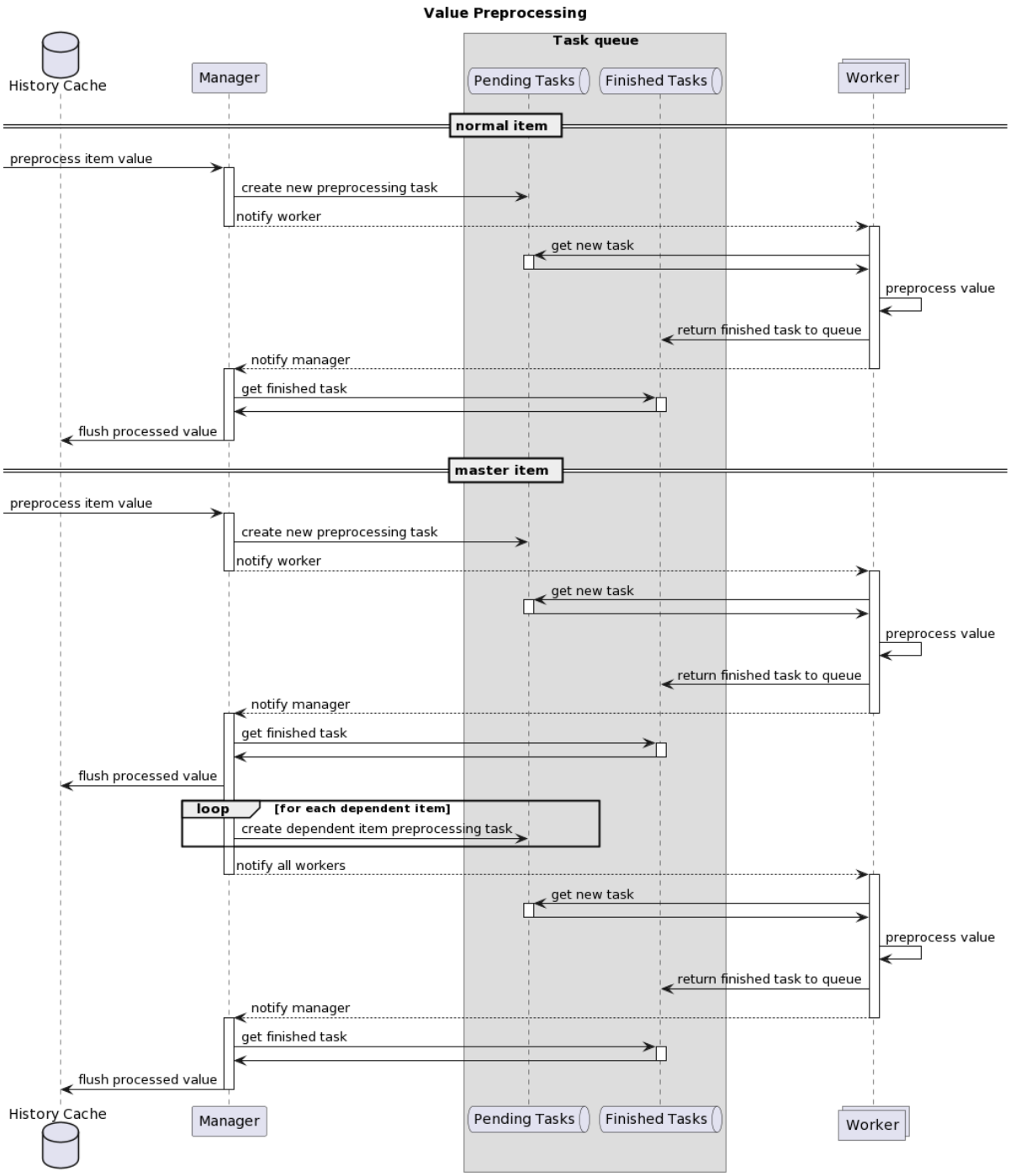
Note:
Элемент данных может изменить свое состояние на NOT SUPPORTED, если нормализация данных завершится с ошибкой (например, когда текстовое значение не может быть преобразовано в число).

- Собранные данные обрабатываются — проверяются триггеры, обновляется конфигурация элемента данных, если элемент данных становится NOT SUPPORTED, и т. д.
- Это считается завершением потока данных с точки зрения обработки значений элементов данных.

Предобработка значения элемента данных

Предобработка данных выполняется в следующих шагах:

- Если у элемента данных нет ни предобработки, ни зависимых элементов данных, его значение либо добавляется в кэш истории, либо отправляется в менеджер LLD. В противном случае значение элемента данных передается в менеджер предобработки с использованием механизма IPC на основе UNIX-сокета.
- Создается задача предобработки, она добавляется в очередь, а рабочие процессы предобработки уведомляются о новой задаче.
- На этом этапе поток данных останавливается до тех пор, пока не появится хотя бы один свободный (то есть не выполняющий никаких задач) рабочий процесс предобработки.
- Когда рабочий процесс предобработки становится доступен, он берет следующую задачу из очереди.
- После завершения предобработки (как при неудачном, так и при успешном выполнении шагов предобработки) предварительно обработанное значение добавляется в очередь завершенных задач, а менеджер уведомляется о новой завершенной задаче.
- Менеджер предобработки преобразует результат в нужный формат (определяемый типом значения элемента данных) и либо добавляет его в кэш истории, либо отправляет в менеджер LLD.
- Если для обработанного элемента данных есть зависимые элементы данных, то они добавляются в очередь предобработки с предварительно обработанным значением основного элемента данных. Зависимые элементы данных помещаются в очередь в обход обычных запросов на предобработку значений, но только для основных элементов данных со значением, которое задано, и не находящихся в состоянии NOT SUPPORTED.



Обратите внимание, что на диаграмме предобработка основного элемента данных немного упрощена за счет пропуска кэширования предобработки.

Очередь предварительной обработки

Очередь предварительной обработки организована следующим образом:

- список ожидающих задач:
  - задачи, созданные непосредственно из запросов предварительной обработки значений в порядке их получения
- список немедленных задач (обрабатываемых до ожидающих задач):
  - задачи тестирования (созданы в ответ на запросы тестирования элементов данных/предварительной обработки веб-интерфейсом)
  - задачи зависимых элементов данных
  - последовательные задачи (задачи, которые должны выполняться в строгом порядке):
  - имеющие шаги предварительной обработки с использованием последнего значения:
    - изменение
    - троттлинг
    - JavaScript (кэширование байт-кода)

    - кэширование предварительной обработки зависимых элементов данных
- список завершённых задач

Кэширование предварительной обработки

Кэширование предварительной обработки было введено для повышения производительности предварительной обработки для нескольких зависимых элементов данных, имеющих схожие шаги предварительной обработки (что является распространённым результатом LLD).

Кэширование выполняется путём предварительной обработки одного зависимого элемента данных и повторного использования некоторых внутренних данных предварительной обработки для остальных зависимых элементов данных. Кэширование предварительной обработки поддерживается только для первого шага предварительной обработки следующих типов:

- шаблон Prometheus (индексирует входные данные по метрикам)
- JSONPath (разбирает данные в дерево объектов и индексирует первое выражение `[?(@.path == "     ")]`)

Рабочие процессы предварительной обработки

Файл конфигурации сервера Zabbix позволяет пользователям задавать количество потоков рабочих процессов предварительной обработки. Параметр конфигурации StartPreprocessors следует использовать для задания числа заранее запущенных экземпляров рабочих процессов предварительной обработки, которое должно как минимум соответствовать числу доступных ядер CPU.

Если задачи предварительной обработки не ограничены CPU и предполагают частые сетевые запросы, рекомендуется настроить дополнительных рабочих процессов. Оптимальное количество рабочих процессов предварительной обработки можно определить по многим факторам, включая количество "preprocessable" элементов данных (элементов данных, для которых требуется выполнение любых шагов предварительной обработки), количество процессов сбора данных, среднее число шагов предварительной обработки для элемента данных и т. д. Недостаточное количество рабочих процессов может привести к высокому потреблению памяти. Для устранения неполадок, связанных с чрезмерным использованием памяти в вашей установке Zabbix, см. Профилирование чрезмерного использования памяти с tcmalloc.

Однако если предположить, что нет ресурсоемких операций предварительной обработки, таких как разбор больших фрагментов XML/JSON, количество рабочих процессов предварительной обработки может соответствовать общему числу сборщиков данных. В этом случае в основном (за исключением случаев, когда данные от сборщика поступают пакетно) для собранных данных будет как минимум один свободный рабочий процесс предварительной обработки.

Warning:
Слишком большое количество процессов сбора данных (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) вместе с IPMI manager, SNMP trapper и рабочими процессами предварительной обработки может исчерпать лимит файловых дескрипторов на процесс для preprocessing manager.

 Исчерпание лимита файловых дескрипторов на процесс приведет к остановке сервера Zabbix, обычно вскоре после запуска, но иногда это занимает больше времени. Чтобы избежать таких проблем, проверьте файл конфигурации сервера Zabbix и оптимизируйте количество одновременных проверок и процессов. Кроме того, при необходимости убедитесь, что лимит файловых дескрипторов установлен достаточно высоким, проверив и скорректировав системные ограничения.

Поток обработки значения

Предварительная обработка значения элемента данных выполняется в несколько шагов (или фаз) несколькими процессами. Это может привести к тому, что:

- Зависимый элемент данных может получить значения, в то время как основной элемент данных — нет. Такое может произойти при следующем сценарии:
    - У основного элемента данных тип значения `UINT` (можно использовать элемент данных траппер), у зависимого элемента данных тип значения `TEXT`.
    - Шаги предварительной обработки не требуются ни для основного, ни для зависимого элементов данных.
    - Основному элементу данных необходимо передать текстовое значение (например, «abc»).
    - Поскольку нет шагов предварительной обработки для выполнения, менеджер предварительной обработки проверяет, что основной элемент данных не находится в НЕПОДДЕРЖИВАЕМОМ состоянии и что имеется значение (оба условия выполняются), и помещает в очередь зависимый элемент данных с таким же значением, что и основной элемент данных (поскольку шаги предварительной обработки отсутствуют).
    - Когда основной и зависимый элементы данных переходят в фазу синхронизации истории, основной элемент данных становится НЕПОДДЕРЖИВАЕМЫМ из-за ошибки конвертации значения (текстовые данные невозможно преобразовать в целое положительное число).

В результате зависимый элемент данных получает значение, тогда как основной элемент данных меняет свое состояние на НЕ ПОДДЕРЖИВАЕТСЯ.

- Зависимый элемент данных получает значение, которое отсутствует в истории основного элемента данных. Данный случай очень похож на предыдущий, за исключением типа основного элемента данных. Например, если у основного элемента данных используется тип CHAR, то значение основного элемента данных будет усечено на стадии синхронизации истории, в то время как зависимые элементы данных получат свои значения с изначального (не усечённого) значения основного элемента данных.

Примеры предварительной обработки

Обзор

В этом разделе представлены примеры использования шагов предварительной обработки для решения некоторых практических задач.

Фильтрация записей журнала событий VMware

В этом примере используется шаг предварительной обработки Совпадение регулярному выражению (Matches regular expression) для фильтрации ненужных событий из журнала событий VMware.

1. Убедитесь, что на работающем узле сети гипервизора VMware элемент данных vmware.eventlog присутствует и работает должным образом. Обратите внимание, что элемент данных журнала событий уже может присутствовать у гипервизора, если при создании узла сети был присоединён шаблон VMware.

2. На узле сети гипервизора VMware создайте зависимый элемент данных с типом *Журнал (лог) (Log)* и выберите элемент данных журнала событий в качестве основного элемента данных.

3. На вкладке *Предобработка (Preprocessing)* зависимого элемента данных нажмите *Добавить (Add)*, чтобы создать шаг предобработки, и выберите из выпадающего списка *Совпадение регулярному выражению (Matches regular expression)*. Затем укажите один из следующих параметров:

- Для фильтрации всех событий журнала:

```
.* logged in .*
```

- Для фильтрации строк, содержащих имена пользователей после «User»:

```
\bUser\s+\K\S+
```

Attention:
Если нет соответствия регулярному выражению, то зависимый элемент данных станет неподдерживаемым с соответствующим сообщением об ошибке. Чтобы избежать этого, отметьте флажок *Другое при ошибке (Custom on fail)* и выберите опцию — например, отбросить значение либо выставить пользовательское значение. Обратите внимание, что отброшенные значения в базе данных не сохраняются; в результате триггеры не вычисляются и данные о динамике изменений не генерируются.

В качестве альтернативы вы можете использовать шаг предварительной обработки Регулярное выражение для извлечения соответствующих групп и управления выводом:

- Чтобы извлечь и вывести из журнала событий полное событие, содержащее строку «logged in», укажите следующие параметры:

```
(Pattern): .*logged in.*
(Output): \0
```

- Чтобы извлечь и вывести имена пользователей после «User»:

```
(Pattern): User (.*?)(?=\ )
(Output): \1
```

Проверка типа полученного значения

В этом примере используется шаг предварительной обработки Custom multiplier, чтобы проверить, является ли тип полученного значения элемента данных числовым.

На вкладке *Предварительная обработка* элемента данных выберите шаг предварительной обработки *Custom multiplier* и укажите следующий параметр (умножает полученное значение на 1):

```
1
```

Attention:
Если предварительная обработка завершается сбоем (например, входные данные не являются числовыми), элемент данных становится неподдерживаемым с соответствующим сообщением об ошибке. Чтобы избежать этого, установите флажок *Custom on fail* и выберите вариант, например, отклонение значения или задание собственного значения. Обратите внимание, что отклоненные значения не сохраняются в базе данных; в результате триггеры не вычисляются, а данные трендов не формируются.

Проверка на неподдерживаемое значение

В этом примере используется шаг предварительной обработки Проверка на неподдерживаемое значение (Check for not supported value) для проверки того, что не удалось извлечь значение элемента данных.

Когда процесс поллера Zabbix сервера/прокси пытается собрать значение элемента данных, он может:

- Вернуть корректный результат.
- Вернуть результат, который изначально кажется корректным, но может стать неподдерживаемым позже (например, из-за несоответствия типов значения после предобработки).
- Вернуть ошибку получения значения, в результате чего элемент станет неподдерживаемым. Распространённые причины включают:
  - Неизвестный ключ элемента данных (для Zabbix агента, Простой проверки или внутренних элементов данных Zabbix)
  - Неизвестный OID (SNMP агент), неизвестный датчик (IPMI агент) или отсутствие метрики JMX (JMX агент)
  - Невозможность прочитать файл трапов (SNMP trap)
  - Скрипт не найден (Внешняя проверка)
  - Нет такого URL (HTTP агент, Браузер)
  - Неуспешный вход (SSH агент, TELNET агент)
  - Недопустимый синтаксис формулы (Вычисляемый), ошибка синтаксиса JavaScript (Скрипт) или некорректный SQL (Монитор баз данных)

Чтобы обнаружить и обработать ошибки сбора значений элементов данных, вы можете использовать шаг предварительной обработки *Проверка на неподдерживаемое значение (Check for not supported value)*. Обратите внимание, что этот шаг всегда выполняется первым и обнаруживает только ошибки, которые возникают до начала предварительной обработки.

На вкладке *Предобработка (Preprocessing)* элемента данных выберите шаг предобработки *Проверка на неподдерживаемое значение (Check for not supported value)* и укажите один из следующих параметров:

- Для любых ошибок:

```
:          (any error)
```

- Для ошибок, содержащих «cannot connect»:

```
:              (error matches)
: (?i)cannot connect
```

Затем используйте параметр *Другое при ошибке (Custom on fail)*, чтобы отбросить значение (в данном случае ошибку), задать пользовательское значение или вернуть пользовательское сообщение об ошибке. Обратите внимание, что отброшенные значения в базе данных не сохраняются; в результате триггеры не вычисляются и данные о динамике изменений не генерируются.

Функциональность JSONPath

Обзор

В этом разделе описывается поддерживаемый функционал JSONPath в шагах предварительной обработки значений элементов данных.

JSONPath состоит из сегментов, разделенных точками. Сегмент может быть иметь форму либо простого слова, представляющего имя значения JSON, либо подстановочного символа (*), либо более замысловатой конструкции, заключённой в квадратные скобки. Точка перед сегментом скобок необязательна и может быть опущена.

Пример JSONPath	Описание
`$.object.name`	Возвращает содержимое object.name.
`$.object['name']`	Возвращает содержимое object.name.
`$.object.['name']`	Возвращает содержимое object.name.
`$["object"]['name']`	Возвращает содержимое object.name.
`$.['object'].["name"]`	Возвращает содержимое object.name.

Пример JSONPath	Описание
`$.object.history.length()`	Возвращает количество элементов массива object.history.
`$[?(@.name == 'Object')].price.first()`	Возвращает значение свойства `price` первого объекта по имени «Object».
`$[?(@.name == 'Object')].history.first().length()`	Возвращает количество элементов массива history первого объекта с именем «Object».
`$[?(@.price > 10)].length()`	Возвращает количество объектов с price больше чем 10.

Смотрите также: Экранирование специальных символов из значений LLD макросов в JSONPath.

Поддерживаемые сегменты

Сегмент	Описание
`< >`	Соответствие свойству объекта по имени.
`*`	Соответствие всем свойствам объекта.
`['< >']`	Соответствие свойству объекта по имени.
`['< >', '< >', ...]`	Соответствие свойству объекта по любому из перечисленных имён.
`[< >]`	Соответствие элементу массива по его индексу.
`[< >, < >, ...]`	Соответствие элементу массива по любому из перечисленных индексов.
`[*]`	Соответствие всем свойствам объекта или элементам массива.
`[< >:< >]`	Соответствие элементам массива по заданному диапазону: **<начало>** — первый индекс соответствия (включительно). Если не указан, соответствует всем элементам с самого начала. В случае отрицательного значения указывает начальное смещение от конца массива. **<конец>** — последний индекс соответствия (не включая). Если не указан, соответствует всем элементам массива до самого конца. В случае отрицательного значения указывает начальное смещение от конца массива.
`[?(< >)]`	Соответствие объектов / элементов массива с применением выражения фильтра.

Чтобы найти соответствующий сегмент, игнорируя его происхождение (отсоединённый сегмент), его необходимо указывать с префиксом '..', например `$..name` или `$..['name']` вернёт значения всех свойств «name».

Соответствующие имена элементов можно извлечь, добавив суффикс ~ к JSONPath. Он возвращает имя соответствующего объекта или индекс в строковом формате соответствующего элемента массива. Формат вывода следует тем же самым правилам, что и остальные запросы JSONPath — результаты определённого пути возвращаются «как есть», а результаты неопределённого пути возвращаются в виде массива. Однако, нет особого смысла извлекать имя соответствующего элемента определённого пути — оно и так уже известно.

Выражение фильтра

Выражение фильтра — это арифметическое выражение в инфиксной нотации.

Поддерживаемые операнды:

Операнд	Описание

`"<text>"`
`'<text>'`
	Текстовая константа.
	
	Пример:
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	Числовая константа с поддержкой научной нотации. Пример: 123
`<jsonpath starting with $>`	Значение, на которое указывает JSONPath от корневого узла входного документа; поддерживаются только определенные пути. Пример: `$.object.name`

Операнд	Описание
`<jsonpath starting with @>`	Значение, на которое указывает JSONPath от текущего объекта/элемента; поддерживаются только определенные пути. Пример: `@.name`

Поддерживаемые операторы: |Оператор|Тип|Описание|Результат| |-|---|---|---| |−|Binary|Вычитание|Number| |+|Binary|Сложение|Numb |/|Binary|Деление|Number| |*|Binary|Умножение|Number| |==|Binary|Равенство|Boolean (1/0)| |!=|Binary|Неравенство|Boolean (1/0)| |<|Binary|Меньше чем|Boolean (1/0)| |<=|Binary|Меньше или равно|Boolean (1/0)| |>|Binary|Больше чем|Boolean (1/0)| |>=|Binary|Больше или равно|Boolean (1/0)| |=~|Binary|Соответствует регулярному выражению|Boolean (1/0)| |!|Unary|Булево НЕ|Boolean (1/0)| ||||Binary|Булево ИЛИ|Boolean (1/0)| |&&|Binary|Булево И|Boolean (1/0)|

Функции

Функции можно использовать в конце JSONPath. Несколько функций можно включать в цепочку, если предыдущая функция возвращает значение, которое принимается следующей функцией.

Поддерживаемые функции:

Функция	Описание	Вход	Выход
`avg`	Среднее значение из чисел во входящем массиве	Массив чисел	Число
`min`	Минимальное значение из чисел во входящем массиве	Массив чисел	Число
`max`	Максимальное значение из чисел во входящем массиве	Массив чисел	Число
`sum`	Сумма чисел во входящем массиве	Массив чисел	Число
`length`	Количество элементов во входящем массиве	Массив	Число
`first`	Первый элемент массива	Массив	Конструкция JSON (объект, массив, значение), в зависимости от содержимого входящего массива

Функциями агрегации JSONPath принимаются числовые значения в кавычках. Это означает, что значения преобразуются из строкового типа в числовой, если требуется агрегирование.

Несовместимые входные данные повлекут ошибку в функции.

Результирующее значение

JSONPath можно разделить на определённые и неопределённые пути. Определённый путь может вернуть только null или одно совпадение. Неопределённый путь может вернуть несколько совпадений: JSONPath с отсоединёнными несколькими именами / списком индексов, фрагментами массива или сегментами выражения. Однако, когда используется функция, JSONPath становится определённым, так как функции всегда имеют одиночное результирующее значение.

Определённый путь возвращает объект/массив/значение, на которое он ссылается. Напротив, неопределённый путь возвращает массив соответствующих объектов/массивов/значений.

Attention:
Порядок свойств в результатах запроса JSONPath может не соответствовать порядку свойств исходного JSON'а из-за внутренних методов оптимизации. Например, JSONPath `$.books[1]["author", "title"]` может вернуть `["title", "author"]`. Если сохранение исходного порядка свойств имеет важное значение, следует рассмотреть альтернативные методы обработки пост-запроса.

Правила форматирования пути

Пробельные символы (пробелы и символы табуляции) можно свободно использовать в сегментах с представлением в виде квадратных скобок и в выражениях, например: `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Строки должны быть заключены в одинарные (') или двойные (") кавычки. Внутри строк одинарные или двойные кавычки (в зависимости от того, в какие кавычки заключена строка) и обратные косые черты (\) экранируются при помощи символа обратной косой черты (\).

Пример


```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "        ",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "      ",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "     ",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "         ",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "                      ",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "                    A5",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "                ",
      "active": false,
      "methods": [
        {
          "description": "          ",
          "price": 46
        },
        {
          "description": "           ,           ",
          "price": 24.5
        },
        {
          "description": "              ",
```

```
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "   \"filters\""
  },
  "closed message": "        ",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	без ”filters”

`$.filters`	definite	{
		”price”: 10,
		”category”: ”fiction”,
		”no filters”: ”без \”filters\””
		}

`$.books[1].title`	definite	Меч чести
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Моби Дик”, ”Изречения века”]
`$.books[1]['author', "title"]`	indefinite	[”Меч чести”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Меч чести”] Примечание: Этот запрос показывает, что в запросах можно использовать арифметические операции; его можно упростить до `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Меч чести”, ”Властелин колец”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Изречения века”, ”Моби Дик”, ”Властелин колец”]
`$.books[?(@.id != 2)].title`	indefinite	[”Изречения века”, ”Моби Дик”, ”Властелин колец”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Изречения века”, ”Меч чести”, ”Властелин колец”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Властелин колец”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Изречения века”, ”Властелин колец”]

JSONPath	Type	Result
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Меч чести”, ”Властелин колец”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Меч чести”,”Моби Дик”,”Властелин колец”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Моби Дик”]

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Изречения века”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Меч чести”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Моби Дик”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”Властелин колец”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Печать и сборка книги в формате A5”, ”Переплет порванной книги”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Меч чести
`$..tags.first().length()`	definite	5 Примечание: `$..tags` — это неопределенный путь, поэтому он возвращает массив совпавших элементов, то есть `[["a", "b", "c", "d", "e" ]]`; `first()` возвращает первый элемент, то есть `["a", "b", "c", "d", "e"]`; `length()` вычисляет длину элемента, то есть 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Примечание: Запрос без совпадений возвращает NULL для определенных и неопределенных путей.
`$.services[?(@.active=="true")].price.min()`	indefinite	[100,1000] Примечание: При сравнении логических значений необходимо использовать текстовые константы.

JSONPath	Type	Result
$.services[?(@.active=="false")].name	Definite path	[1002]
$.services[?(@.servicegroup=="demo")].name	Definite path	[1002]
$.services[?(@.servicegroup=="demo")]~.first()	Result conversion	Примечание: При сравнении логических значений необходимо использовать текстовые константы.

Экранирование специальных символов в значениях макросов LLD в JSONPath

Когда макросы низкоуровневого обнаружения используются в предварительной обработке JSONPath и их значения подставляются, применяются следующие правила экранирования специальных символов:

- только символ обратной косой черты (\) и двойная кавычка (") учитываются при экранировании;
- если подставленное значение макроса содержит эти символы, каждый из них экранируется обратной косой чертой;
- если они уже экранированы обратной косой чертой, это не считается экранированием, и и обратная косая черта, и следующий за ней специальный символ экранируются еще раз.

Например:

JSONPath	значение макроса LLD	После подстановки
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

При использовании в выражении макрос, который может содержать специальные символы, следует заключать в двойные кавычки:

JSONPath	значение макроса LLD	После подстановки	Результат
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Неверное выражение JSONPath**

При использовании в пути макрос, который может содержать специальные символы, следует заключать в квадратные скобки **и** двойные кавычки:

JSONPath	значение макроса LLD	После подстановки	Результат
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Неверное выражение JSONPath**

Предварительная обработка JavaScript

Обзор

В этом разделе приведены сведения о предварительной обработке при помощи JavaScript.

Attention:
Не используйте необъявленные присваивания в предварительной обработке JavaScript. Используйте `var` для объявления локальных переменных.

Предобработка JavaScript

Предобработка JavaScript выполняется путем вызова функции JavaScript с одним параметром 'value' и телом функции, заданным пользователем. Результатом шага предобработки является значение, возвращаемое этой функцией. Например, чтобы выполнить преобразование из градусов Фаренгейта в градусы Цельсия, введите:

```
return (value - 32)  * 5 / 9
```

в параметрах предобработки JavaScript, которые будут обернуты сервером в функцию JavaScript:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Входной параметр 'value' всегда передается как строка. Возвращаемое значение автоматически приводится к строке с помощью метода toString() (если это не удается, то ошибка возвращается как строковое значение), за несколькими исключениями:

- возврат значения undefined приведет к ошибке;
- возврат значения null приведет к тому, что входное значение будет отброшено, аналогично предобработке 'Discard value' при действии 'Custom on fail'.

Ошибки можно возвращать, выбрасывая значения/объекты (обычно либо строки, либо объекты Error).

Например:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Для каждого скрипта установлен 10-секундный тайм-аут выполнения (в зависимости от скрипта срабатывание тайм-аута может занять больше времени); при его превышении будет возвращена ошибка. Также действует ограничение на объем heap в 512 мегабайт.

Байт-код шага предобработки JavaScript кэшируется и повторно используется при следующем применении шага. Любые изменения в шагах предобработки элемента данных приведут к сбросу кэшированного скрипта и его последующей повторной компиляции.

Последовательные сбои выполнения (3 подряд) приведут к повторной инициализации движка, чтобы снизить вероятность того, что один скрипт нарушит среду выполнения для следующих скриптов (это действие регистрируется с DebugLevel 4 и выше).

Предобработка JavaScript реализована с использованием движка JavaScript Duktape.

> **Note:**
> Встроенные определения методов объектов JavaScript нельзя переопределять в скриптах предобработки элемента данных.

См. также: Additional JavaScript objects and global functions

Использование макросов в скриптах

В JavaScript-коде можно использовать пользовательские макросы (а также LLD-макросы в контексте низкоуровневого обнаружения). Если скрипт содержит пользовательские макросы, эти макросы разрешаются сервером/прокси перед выполнением определенных шагов предварительной обработки. Обратите внимание, что при тестировании шагов предварительной обработки во веб-интерфейсе значения макросов не будут подставлены и их нужно будет ввести вручную.

> **Note:**
> Контекст игнорируется при замене макроса его значением. Значение макроса вставляется в код как есть; невозможно добавить дополнительное экранирование перед помещением значения в JavaScript-код. Имейте в виду, что в некоторых случаях это может привести к ошибкам JavaScript.

В примере ниже, если полученное значение превышает значение макроса {$THRESHOLD}, вместо него будет возвращено значение порога (если оно задано):

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Примеры

Следующие примеры иллюстрируют, как можно использовать предобработку JavaScript.

Каждый пример содержит краткое описание, тело функции для параметров предварительной обработки JavaScript и результат шага предварительной обработки — значение, возвращаемое функцией.

Пример 1: Преобразование числа (из научной нотации в целое число)

Преобразовать число «2.62128e+07» из научной нотации в целое число.

```
return (Number(value))
```

Значение, возвращаемое функцией: 26212800.

Пример 2: Преобразование числа (двоичного в десятичное)

Преобразовать двоичное число «11010010» в десятичное.

```
return(parseInt(value,2))
```

Значение, возвращаемое функцией: 210.

Пример 3: Округление числа

Округлить число «18.2345» до двух цифр.

```
return(Math.round(value* 100) / 100)
```

Значение, возвращаемое функцией: 18.23.

Пример 4: Подсчёт букв в строке

Подсчитать количество букв в строке «Zabbix».

```
return (value.length)
```

Значение, возвращаемое функцией: 6.

Пример 5: Получить оставшееся время

Получить время (в секундах), оставшееся до момента истечения срока действия сертификата (Feb 12 12:33:56 2022 GMT).

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();

return parseInt((Date.parse(ISOdate) - now) / 1000);
```

Значение, возвращаемое функцией: 44380233.

Пример 6: Удаление свойств JSON

Изменить структуру данных JSON, удалив все свойства с ключом "data_size" или "index_size".

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

Значение, принимаемое функцией:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

Значение, возвращаемое функцией:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Пример 7: Преобразование статуса Apache в JSON

Преобразовать значение, полученное от элемента данных агента Zabbix web.page.get (например, web.page.get[http://127.0.0.1:80/server-status?auto]), в объект JSON.

```
//             Apache   JSON

//
var lines = value.split('\n');

//              «output»
var output = {};

//          «workers»
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

//                  «lines»       «output»         (     -     )
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);

    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

//
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

//           «Scoreboard»                worker-
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

//            worker-        «output»
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

//          JSON
return JSON.stringify(output);
```

Значение, принятое функцией:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 11:08:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 405
Content-Type: text/plain; charset=ISO-8859-1

127.0.0.1
ServerVersion: Apache/2.4.52 (Ubuntu)
ServerMPM: prefork
Server Built: 2023-03-08T17:32:01
CurrentTime: Monday, 27-Mar-2023 14:08:39 EEST
RestartTime: Monday, 27-Mar-2023 12:19:59 EEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 6520
ServerUptime: 1 hour 48 minutes 40 seconds
Load1: 0.56
Load5: 0.33
Load15: 0.28
Total Accesses: 2476
Total kBytes: 8370
Total Duration: 52718
CPUUser: 8.16
CPUSystem: 3.44
CPUChildrenUser: 0
CPUChildrenSystem: 0
CPULoad: .177914
Uptime: 6520
ReqPerSec: .379755
BytesPerSec: 3461.58
BytesPerReq: 3461.58
DurationPerReq: 21.2916
BusyWorkers: 2
IdleWorkers: 6
Scoreboard: ____KW__.....................................................................................
```

Значение, возвращаемое функцией:


```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
```

```
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................
    "Workers": {
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
        "cleanup": 0,
        "slot": 142
    }
}
```

Дополнительные объекты JavaScript

Обзор

Этот раздел описывает дополнения Zabbix к языку JavaScript, который реализован при помощи Duktape, и поддерживаемые глобальные функции JavaScript.

Attention:
Не используйте необъявленные присваивания в предварительной обработке JavaScript. Используйте `var` для объявления локальных переменных.

Встроенные объекты

Zabbix

Объект Zabbix даёт возможность взаимодействия с внутренним функционалом Zabbix.

Метод	Описание
`log( _ , )`	Записывает <сообщение> в журнал Zabbix с использованием уровня журнала <уровень_журнала> (смотрите параметр DebugLevel в файле конфигурации).

Пример:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

Вы можете использовать следующие псевдонимы:

Псевдоним	Псевдоним к
console.log(объект)	Zabbix.log(4, JSON.stringify(объект))
console.warn(объект)	Zabbix.log(3, JSON.stringify(объект))
console.error(объект)	Zabbix.log(2, JSON.stringify(объект))

Attention:
Суммарный размер всех записываемых в журнал сообщений ограничен величиной 8 мегабайт на выполняемый скрипт.

Метод	Описание
sleep( )	Задержка выполнения JavaScript на миллисекунд.

Пример (задержка выполнения на 15 секунд):

```
Zabbix.sleep(15000)
```

HttpRequest

Этот объект инкапсулирует дескриптор cURL и позволяет выполнять простые HTTP-запросы. Ошибки вызываются как исключения.

Attention:
Инициализация нескольких объектов HttpRequest ограничена 10 объектами за одно выполнение скрипта.

Method	Description
addHeader(value)	Добавляет поле HTTP-заголовка. Это поле используется для всех последующих запросов, пока не будет очищено с помощью метода clearHeader(). Общая длина полей заголовка, которые можно добавить в один объект HttpRequest, ограничена 128 Кбайт (включая специальные символы и имена заголовков).
clearHeader()	Очищает HTTP-заголовок. Если поля заголовка не заданы, HttpRequest установит Content-Type в application/json, если отправляемые данные имеют формат JSON; в противном случае будет установлено значение text/plain.
connect(url)	Отправляет HTTP-запрос CONNECT по указанному URL и возвращает ответ.
customRequest(method, url, data)	Позволяет указать любой HTTP-метод в первом параметре. Отправляет запрос указанного метода по URL с необязательной полезной нагрузкой *data* и возвращает ответ.
delete(url, data)	Отправляет HTTP-запрос DELETE по URL с необязательной полезной нагрузкой *data* и возвращает ответ.
getHeaders(<asArray>)	Возвращает объект полученных полей HTTP-заголовка. Параметру asArray можно задать значение ”true” (например, getHeaders(true)), ”falseили оставить его неопределенным. Если задано значение ”true”, значения полученных полей заголовка возвращаются в виде массивов; этот вариант следует использовать для получения значений полей нескольких заголовков с одинаковыми именами. Если параметр не задан или задано значение ”false”, значения полученных полей заголовка возвращаются в виде строк.||get(url, data)| HTTP- GET URL *data* .| |head(url)| HTTP- HEAD URL .| |options(url)| HTTP- OPTIONS URL .| |patch(url, data)| HTTP- PATCH URL *data* .| |put(url, data)| HTTP- PUT URL *data* .| |post(url, data)| HTTP- POST URL *data* .| |getStatus()| HTTP- .| |setProxy(proxy)| HTTP- "proxy". , .| |setHttpAuth(bitmask, username, password)|Устанавливает разрешенные методы HTTP-аутентификации (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) в параметре ’bitmask’. Флаг HTTPAUTH_NONE позволяет отключить HTTP-аутентификацию. Примеры: “request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)“ request.setHttpAuth(HTTPAUTH_NONE)| |trace(url, data)‘

Пример:

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);
```

```
    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

Объект XML позволяет обрабатывать XML данные в элементах данных и в предварительной обработке низкоуровневых обнаружений, а также в вебхуках.

Attention:
Чтобы использовать объект XML, сервер/прокси должны быть скомпилированы с поддержкой libxml2.

Метод	Описание
XML.query(,)	Получение содержимого узла с использованием XPath. Возвращает null, если узел не найден. **выражение** — выражение XPath; **данные** — данные XML в виде строки.
XML.toJson( )	Преобразование данных в XML формате в JSON.
XML.fromJson( )	Преобразование данных в JSON формате в XML.

Пример:

Входящие данные:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
```

```
        <calories>650</calories>
    </food>
</menu>
```

Вывод:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Правила сериализации

Преобразование XML в JSON обрабатывается в соответствии со следующими правилами (для преобразования JSON в XML применяются обратные правила):

1. Атрибуты XML преобразуются в ключи, имеющие в своих именах префикс '@'.

Пример:

Входящие данные:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Вывод:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Элементы, которые закрываются в самих себе (<foo/>), будут преобразованы как имеющие значение «null».

Пример:

Входящие данные:

```
<xml>
  <foo/>
</xml>
```

Вывод:

```
{
  "xml": {
    "foo": null
  }
}
```

3. Пустые атрибуты (со значением "") будут преобразованы как имеющие значение пустой строки ('').

Пример:

Входящие данные:

```xml
<xml>
  <foo bar="" />
</xml>
```

Вывод:

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Несколько дочерних узлов с идентичными именами элементов будут преобразованы в один ключ, который имеет массив значений в качестве своего значения.

Пример:

Входящие данные:

```xml
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Вывод:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Если текстовый элемент не имеет атрибутов и дочерних элементов, он будет преобразован в строку.

Пример:

Входящие данные:

```xml
<xml>
    <foo>BAZ</foo>
</xml>
```

Вывод:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Если текстовый элемент не имеет дочерних элементов, но имеет атрибуты, то текстовое содержимое будет преобразовано в элемент с ключом «#text» и содержимым в виде значения; атрибуты будут преобразованы в соответствии с правилом сериализации 1.

Пример:

Входящие данные:

```xml
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Вывод:

```
{
  "xml": {
    "foo": {
```

```
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Глобальные функции JavaScript

В Duktape реализованы дополнительные глобальные функции JavaScript:

- `btoa(data)` - кодирует данные в строку base64.
- `atob(base64_string)` - декодирует строку base64 в буфер Uint8Array.

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    //          ,                                    .
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- `md5(data)` - вычисляет MD5-хэш данных.
- `sha256(data)` - вычисляет SHA256-хэш данных.
- `hmac('<hash type>',key,data)` - возвращает HMAC-хэш в виде строки в шестнадцатеричном формате; в качестве `hash type` поддерживаются `md5` и `sha256`; параметры `key` и `data` поддерживают двоичные данные.

  Примеры:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` - возвращает вычисленную подпись (RSA-подпись с SHA-256) в виде строки, где:
 hash - допускается только `sha256`, в противном случае будет выдана ошибка.
 key - закрытый ключ. Он должен соответствовать стандарту PKCS#1 или PKCS#8. Ключ может быть предоставлен в разных формах:

  - с пробелами вместо символов новой строки
  - с экранированными или неэкранированными `\n` вместо символов новой строки
  - без каких-либо символов новой строки, в виде строки в одну строку
  - в виде строки в формате JSON

  Ключ также может быть загружен из пользовательского макроса/секретного макроса/vault.

  data - данные, которые будут подписаны. Это может быть строка (также поддерживаются двоичные данные) или буфер (Uint8Array/ArrayBuffer).

  Пример:

  - `sign('sha256',key,data)`

  Для вычисления подписей используется OpenSSL или GnuTLS. Если Zabbix был собран без какой-либо из этих криптографических библиотек, будет выдана ошибка ('missing OpenSSL or GnuTLS library').

Объекты JavaScript элемента данных браузера

Обзор

В этом разделе описываются расширения Zabbix к языку JavaScript, реализованные с помощью Duktape для использования в скрипте браузерного элемента данных. Эти расширения дополняют объекты JavaScript, описанные на странице Дополнительные объекты JavaScript.

Browser

Объект `Browser` управляет сеансами WebDriver, инициализируя сеанс при создании и завершая его при уничтожении. Один скрипт может поддерживать до четырех объектов `Browser`.

Для создания объекта `Browser` используйте синтаксис `new Browser(options)`. Параметр `options` (*JSON object*) задает параметры браузера, обычно это результат метода параметров WebDriver (например, `Browser.chromeOptions()`).

С объектом `Browser` поддерживаются следующие методы.

Method	Description
`navigate(url)`	Перейти по указанному URL. Parameters: `url` - (string) URL, по которому нужно перейти.
`getUrl()`	Возвращает строку с URL открытой страницы.
`getPageSource()`	Возвращает строку с исходным кодом открытой страницы.
`findElement(strategy, selector)`	Возвращает объект `Element` с одним элементом на открытой странице (или возвращает `null`, если ни один элемент не соответствует `strategy` и `selector`). Parameters: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; `selector` - (string) Селектор элемента с использованием указанной стратегии поиска.
`findElements(strategy, target)`	Возвращает массив объектов `Element` с несколькими элементами на открытой странице (или возвращает пустой массив, если ни один элемент не соответствует стратегии поиска и `target`). Parameters: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; `target` - (string) Селектор элемента с использованием указанной стратегии поиска.
`getCookies()`	Возвращает массив объектов `Cookie`.
`addCookie(cookie)`	Установить cookies. Parameters: `cookie` - (`Cookie` object) Cookie для установки.
`getScreenshot()`	Возвращает строку в кодировке base64, представляющую изображение области просмотра браузера.
`setScreenSize(x,y)`	Установить размер области просмотра браузера. Parameters: `x` - (string) Ширина в пикселях; `y` - (string) Высота в пикселях.
`setScriptTimeout(timeout)`	Установить тайм-аут загрузки скрипта. Parameters: `timeout` - (integer) Значение тайм-аута в миллисекундах.
`setSessionTimeout(timeout)`	Установить тайм-аут сеанса (загрузки страницы). Parameters: `timeout` - (integer) Значение тайм-аута в миллисекундах.
`setElementWaitTimeout(timeout)`	Установить тайм-аут стратегии поиска элемента (неявный). Parameters: `timeout` - (integer) Значение тайм-аута в миллисекундах.
`collectPerfEntries(mark)`	Собрать записи производительности для получения с помощью метода `getResult()`. Parameters: `mark` - (string, optional) метка снимка производительности.
`getRawPerfEntries()`	Возвращает массив объектов записей производительности.
`getResult()`	Возвращает объект `Result` со статистикой сеанса браузера (информация об ошибках, снимки производительности и т. д.).

Method	Description
`getError()`	Возвращает объект `BrowserError` с ошибками браузера (или возвращает `null`, если ошибок браузера нет).
`setError(message)`	Установить собственное сообщение об ошибке, которое будет включено в объект `Result`. Parameters: `message` - (string) Сообщение об ошибке.
`discardError()`	Отменить ошибку, которая будет возвращена в объекте `Result`.
`getAlert()`	Возвращает объект `Alert` с предупреждениями браузера (или возвращает `null`, если предупреждений браузера нет).
`chromeOptions()`	Возвращает объект `chromeOptions` с предопределенными параметрами браузера Chrome.
`firefoxOptions()`	Возвращает объект `firefoxOptions` с предопределенными параметрами браузера Firefox.
`safariOptions()`	Возвращает объект `safariOptions` с предопределенными параметрами браузера Safari.
`edgeOptions()`	Возвращает объект `edgeOptions` с предопределенными параметрами браузера Edge.
`switchFrame(target)`	Переключиться на указанный фрейм. Parameters: `target` - (browser element or integer, optional) Целевой фрейм. Чтобы выбрать фрейм по элементу, передайте элемент. Чтобы выбрать фрейм по индексу, передайте число. Если оставить пустым, будет выполнено переключение на контекст просмотра верхнего уровня.

Все методы Browser могут вызывать следующие ошибки:

- `BrowserError` - производная от объекта `Error`, которая возникает, если конструктор `Browser` завершается с ошибкой; содержит дополнительное свойство `browser` с объектом `Browser`, который вызвал эту ошибку `BrowserError`.
- `WebdriverError` - производная от `BrowserError`; содержит те же свойства, что и объект `BrowserError`, и указывает, была ли ошибка сгенерирована в ответ на ошибку в ответе WebDriver.

Element

Объект `Element` возвращается методами `findElement()`/`findElements()` объекта `Browser` и не может быть создан напрямую.

Объект `Element` представляет элемент на веб-странице и предоставляет методы для взаимодействия с ним.

С объектом `Element` поддерживаются следующие методы.

Method	Description
`getAttribute(name)`	Возвращает строковое значение атрибута элемента (или `null`, если указанный атрибут не найден). Parameters: `name` - (string) Имя атрибута.
`getProperty(name)`	Возвращает строковое значение свойства элемента (или `null`, если указанное свойство не найдено). Parameters: `name` - (string) Имя свойства.
`getText()`	Возвращает строковое значение текста элемента.
`click()`	Щелкает по элементу.
`clear()`	Очищает содержимое редактируемого элемента.
`sendKeys(keys)`	Отправляет нажатия клавиш. Parameters: `keys` - (string) Клавиши для отправки.

Cookie

Объект `Cookie` возвращается методом `getCookies()` объекта `Browser` и передаётся в метод `addCookie()`.

Хотя у объекта `Cookie` нет методов, он может содержать следующие свойства:

Свойство	Тип	Описание
name	строка	Имя cookie.
value	строка	Значение cookie.
path	строка	Путь, для которого cookie действителен. По умолчанию `"/"`, если не указан при добавлении cookie.
domain	строка	Домен, в котором cookie является видимым. По умолчанию — домен URL активного документа текущего контекста просмотра сессии, если не указан при добавлении cookie
secure	логический	Указывает, является ли cookie безопасным. По умолчанию `false`, если не указан при добавлении cookie.
httpOnly	логический	Указывает, является ли cookie HTTP-only. По умолчанию `false`, если не указан при добавлении cookie.
expiry	целое число	Время истечения срока действия cookie (в секундах с начала эпохи Unix). Не должно быть установлено, если не указано при добавлении файла cookie.
sameSite	строка	Атрибут `sameSite` файла cookie, который управляет тем, должен ли файл cookie быть ограничен контекстом первой стороны (first-party) или того же сайта. Может быть установлено как `"Lax"` или `"Strict"`. По умолчанию `"None"`, если не указан при добавлении cookie.

Alert

Объект `Alert` представляет оповещение (alert) веб-страницы, возвращается методом `getAlert()` объекта `Browser` и не может быть создан напрямую.

Объект `Alert` содержит свойство `text` с текстом оповещения (или `null`, если оповещения отсутствуют).

Объектом `Alert` поддерживаются следующие методы.

Метод	Описание
`accept()`	Принять оповещение.
`dismiss()`	Отклонить оповещение.

Result

Объект `Result` содержит статистику сессии и возвращается методом `getResult()` объекта `Browser`.

Обычно объект `Result` преобразуется в строку и возвращается из скрипта, а затем разбирается на значения зависимых элементов данных посредством предварительной обработки.

Хотя объект `Result` не имеет методов, он может содержать следующие свойства.

Свойство	Тип	Описание
duration	строка	Длительность сессии от создания сессии до получения результата.
error	объект	Информация об ошибке.
http_status	целое число	Статус HTTP, возвращаемый WebDriver'ом (или 0, если ошибок WebDriver'а нет).
error_code	строка	Ошибка, возвращаемая WebDriver'ом (или пустая строка, если ошибок WebDriver'а нет).
message	строка	Сообщение об ошибке WebDriver'а (или пустая строка, если ошибок WebDriver'а нет).
performance_data	объект	Статистика производительности.
summary	объект	Сводка производительности.
navigation	объект	Сводка навигации.
resource	объект	Сводка ресурсов.

Свойство		Тип	Описание
details		массив объектов	Статистика производительности после каждой операции, которая могла привести к навигации.
	mark	строка	*(опционально)* Отметка снимка производительности, указанная с помощью метода `collectPerfEntries()`.
	navigation	объект	Статистика навигации.
	resource	объект	Сводка ресурсов для этого шага.
	user	массив объектов	Массив статистики отметки / типа измерения.
marks		массив объектов	Отмеченные индексы моментальных снимков производительности.
	name	строка	Имя отметки моментального снимка производительности.
	index	целое число	Индекс моментального снимка производительности в массиве details.

Предобработка CSV в JSON

Обзор

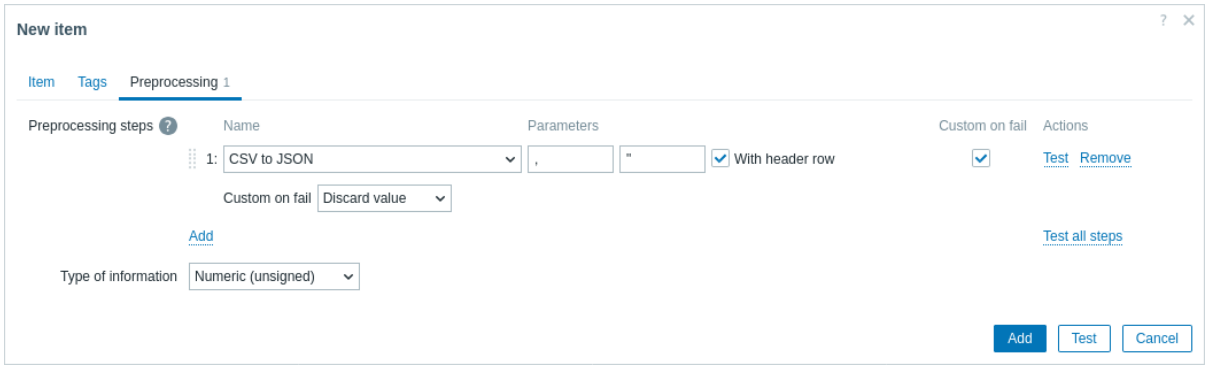
На этом этапе предварительной обработки можно преобразовать данные CSV-файла в формат JSON. Поддерживается для:

- элементов данных (прототипов элементов данных)
- правил обнаружения низкоуровневого обнаружения

Настройка

Чтобы настроить шаг предварительной обработки CSV в JSON:

1. Перейдите на вкладку *Предварительная обработка* в настройках элемента данных/правила обнаружения.
2. Нажмите *Добавить*.
3. Выберите вариант *CSV в JSON*.



Первый параметр позволяет задать пользовательский разделитель. Обратите внимание: если первая строка входных данных CSV начинается с «Sep=» и после нее следует один символ UTF-8, этот символ будет использоваться в качестве разделителя, если первый параметр не задан. Если первый параметр не задан и разделитель не найден в строке «Sep=», в качестве разделителя используется запятая.

Второй необязательный параметр позволяет задать символ кавычек.

Если установлен флажок *С заголовком*, значения строки заголовка будут интерпретироваться как имена столбцов (дополнительную информацию см. в разделе Обработка заголовка).

Если установлен флажок *Пользовательское действие при ошибке*, элемент данных не перейдет в состояние «не поддерживается» в случае сбоя шага предварительной обработки. Кроме того, можно задать пользовательские параметры обработки ошибок: отбросить значение, установить указанное значение или задать указанное сообщение об ошибке.

Обработка заголовка

Строка заголовка CSV файла может обрабатываться двумя разными способами:

- Если флажок *Со строкой заголовка* отмечен — значения строки заголовка интерпретируются как имена столбцов. В этом случае имена столбцов должны быть уникальны и строки с данными не должны содержать больше столбцов, чем строка заголовка;
- Если флажок *Со строкой заголовка* не отмечен — строка заголовка интерпретируется как данные. Имена столбцов генерируются автоматически (1,2,3,4...)

Пример CSV файла:

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:
Знак кавычек внутри заключённого в кавычки поля во входящих данных должен быть экранирован при помощи другого символа кавычек.

Обработка строки заголовка

Вывод JSON, когда ожидается строка заголовка:

```
[
   {
      "Nr":"1",
      "Item name":"active agent item",
      "Key":"agent.hostname",
      "Qty":"33"
   },
   {
      "Nr":"2",
      "Item name":"passive agent item",
      "Key":"agent.version",
      "Qty":"44"
   },
   {
      "Nr":"3",
      "Item name":"active,passive agent items",
      "Key":"agent.ping",
      "Qty":"55"
   }
]
```

Обработка без наличия строки заголовка

Вывод JSON, когда не ожидается строка заголовка:

```
[
   {
      "1":"Nr",
      "2":"Item name",
      "3":"Key"
      "4":"Qty"
   },
   {
      "1":"1",
      "2":"active agent item",
      "3":"agent.hostname"
      "4":"33"
   },
   {
      "1":"2",
      "2":"passive agent item",
      "3":"agent.version"
      "4":"44"
   },
   {
      "1":"3",
      "2":"active,passive agent items",
      "3":"agent.ping"
      "4":"55"
   }
]
```

Типы элементов данных

Обзор

Типы элементов данных охватывают различные методы получения данных из вашей системы. Каждый тип элемента данных имеет собственный набор поддерживаемых ключей элементов данных и обязательных параметров.

В настоящее время Zabbix предлагает следующие типы элементов данных:

- Агент Zabbix
  - Мониторинг файлов журналов
- Простая проверка
  - Мониторинг VMware
- Агент SNMP
- SNMP trap
- Внутренний элемент Zabbix
- Траппер Zabbix
- Внешняя проверка
- Монитор базы данных
- HTTP-агент
  - Проверка Prometheus
- Агент IPMI
- SSH-агент
- Telnet-агент
- JMX-агент
- Вычисляемый элемент данных
- Зависимый элемент данных
- Элемент данных типа «Скрипт»
- Элемент данных типа «Браузер»
- Запрос телеметрии

Подробная информация обо всех типах элементов данных приведена на вложенных страницах этого раздела. Несмотря на то что типы элементов данных предоставляют множество возможностей для сбора данных, дополнительные возможности доступны через пользовательские параметры или загружаемые модули.

Некоторые проверки выполняются только сервером Zabbix (как мониторинг без агента), тогда как для других требуется агент Zabbix или даже Java gateway Zabbix (при мониторинге JMX).

Attention:
Если для определенного типа элемента данных требуется определенный интерфейс (например, для проверки IPMI на узле сети необходим интерфейс IPMI), этот интерфейс должен существовать в определении узла сети.

В определении узла сети можно настроить несколько интерфейсов: агент Zabbix, агент SNMP, JMX и IPMI. Если элемент данных может использовать более одного интерфейса, он будет искать среди доступных интерфейсов узла сети (в следующем порядке: Agent→SNMP→JMX→IPMI) первый подходящий интерфейс, с которым его можно связать.

Все элементы данных, возвращающие текст (типы информации «Символ», «Журнал» и «Текст»), также могут возвращать только пробельные символы (если применимо), устанавливая возвращаемое значение в пустую строку (поддерживается начиная с версии 2.0).

Zabbix агент

Обзор

В этом разделе приведены сведения о ключах элементов данных, которые используют взаимодействие с агентом Zabbix для сбора данных.

Существуют пассивные и активные проверки агента. При настройке элемента данных вы можете выбрать нужный тип:

- *Агент Zabbix* - для пассивных проверок
- *Агент Zabbix (активный)* - для активных проверок

Обратите внимание, что все ключи элементов данных, поддерживаемые агентом Zabbix, также поддерживаются агентом Zabbix 2 нового поколения. См. дополнительные ключи элементов данных, которые можно использовать только с агентом 2.

Поддерживаемые ключи элементов данных

Ниже перечислены ключи элементов данных, которые можно использовать с агентом Zabbix.

Ключи элементов данных перечислены без параметров и дополнительной информации.
Нажмите ключ элемента данных, чтобы просмотреть полную информацию.

Ключ элемента данных	Описание	Группа элементов данных
kernel.maxfiles	Максимальное количество открытых файлов, поддерживаемое ОС.	Ядро
kernel.maxproc	Максимальное количество процессов, поддерживаемое ОС.	
kernel.openfiles	Количество открытых в данный момент файловых дескрипторов.	
log	Мониторинг файла журнала.	Мониторинг журналов
log.count	Количество совпавших строк в отслеживаемом файле журнала.	
logrt	Мониторинг ротируемого файла журнала.	
logrt.count	Количество совпавших строк в ротируемом отслеживаемом файле журнала.	
modbus.get	Чтение данных Modbus.	Modbus
net.dns	Проверка состояния службы DNS.	Сеть
net.dns.perf	Проверка производительности службы DNS.	
net.dns.record	Выполнение DNS-запроса.	
net.if.collisions	Количество коллизий вне окна.	
net.if.discovery	Список сетевых интерфейсов.	
net.if.get	Список сетевых интерфейсов с подробной информацией.	
net.if.in	Статистика входящего трафика сетевого интерфейса.	
net.if.out	Статистика исходящего трафика сетевого интерфейса.	
net.if.total	Суммарная статистика входящего и исходящего трафика сетевого интерфейса.	
net.tcp.listen	Проверка того, находится ли этот TCP-порт в состоянии LISTEN.	
net.tcp.port	Проверка возможности установить TCP-соединение с указанным портом.	
net.tcp.service	Проверка того, запущена ли служба и принимает ли она TCP-соединения.	
net.tcp.service.perf	Проверка производительности TCP-службы.	
net.tcp.socket.count	Возвращает количество TCP-сокетов, соответствующих параметрам.	
net.udp.listen	Проверка того, находится ли этот UDP-порт в состоянии LISTEN.	
net.udp.service	Проверка того, запущена ли служба и отвечает ли она на UDP-запросы.	
net.udp.service.perf	Проверка производительности UDP-службы.	
net.udp.socket.count	Возвращает количество UDP-сокетов, соответствующих параметрам.	
proc.cpu.util	Процент использования процессором процесса.	Процессы
proc.get	Список процессов ОС и их параметров.	
proc.mem	Объем памяти, используемой процессом, в байтах.	
proc.num	Количество процессов.	
sensor	Показания аппаратного датчика.	Датчики
system.boottime	Время загрузки системы.	Система
system.cpu.discovery	Список обнаруженных ЦП/ядер ЦП.	
system.cpu.intr	Прерывания устройств.	
system.cpu.load	Загрузка ЦП.	
system.cpu.num	Количество ЦП.	
system.cpu.switches	Количество переключений контекста.	
system.cpu.util	Процент использования ЦП.	
system.hostname	Имя узла сети системы.	
system.hw.chassis	Информация о корпусе.	
system.hw.cpu	Информация о ЦП.	
system.hw.devices	Список устройств PCI или USB.	
system.hw.macaddr	Список MAC-адресов.	
system.localtime	Системное время.	
system.run	Выполнение указанной команды на узле сети.	
system.stat	Статистика системы.	
system.sw.arch	Информация об архитектуре программного обеспечения.	
system.sw.os	Информация об операционной системе.	
system.sw.os.get	Подробная информация об операционной системе (версия, тип, название дистрибутива, младшая и старшая версии и т. д.).	
system.sw.packages	Список установленных пакетов.	
system.sw.packages.get	Подробный список установленных пакетов.	

Ключ элемента данных	Описание	Группа элементов данных
system.swap.in	Статистика загрузки данных в пространство подкачки (с устройства в память).	
system.swap.out	Статистика выгрузки данных из пространства подкачки (из памяти на устройство).	
system.swap.size	Размер пространства подкачки в байтах или в процентах от общего объема.	
system.uname	Идентификация системы.	
system.uptime	Время работы системы в секундах.	
system.users.num	Количество вошедших в систему пользователей.	
vfs.dev.discovery	Список блочных устройств и их типов.	Виртуальные файловые системы
vfs.dev.get	Список блочных устройств с подробной информацией.	
vfs.dev.read	Статистика чтения с диска.	
vfs.dev.write	Статистика записи на диск.	
vfs.dir.count	Количество записей в каталоге.	
vfs.dir.get	Список записей каталога.	
vfs.dir.size	Размер каталога.	
vfs.file.cksum	Контрольная сумма файла, вычисленная с помощью алгоритма UNIX cksum.	
vfs.file.contents	Получение содержимого файла.	
vfs.file.exists	Проверка существования файла.	
vfs.file.get	Возвращает информацию о файле.	
vfs.file.md5sum	Контрольная сумма файла MD5.	
vfs.file.owner	Получение владельца файла.	
vfs.file.permissions	Возвращает строку из 4 цифр, содержащую восьмеричное число с разрешениями UNIX.	
vfs.file.regexp	Получение строки из файла.	
vfs.file.regmatch	Поиск строки в файле.	
vfs.file.size	Размер файла.	
vfs.file.time	Информация о времени файла.	
vfs.fs.discovery	Список смонтированных файловых систем с их типами и параметрами монтирования.	
vfs.fs.get	Список смонтированных файловых систем с их типами и параметрами монтирования, а в режиме *full* — доступное дисковое пространство и статистика inode.	
vfs.fs.inode	Количество или процент inode.	
vfs.fs.size	Дисковое пространство в байтах или в процентах от общего объема.	
vm.memory.size	Объем памяти в байтах или в процентах от общего объема.	Виртуальная память
web.page.get	Получение содержимого веб-страницы.	Веб-мониторинг
web.page.perf	Время загрузки полной веб-страницы.	
web.page.regexp	Поиск строки на веб-странице.	
agent.hostmetadata	Метаданные узла сети агента.	Zabbix
agent.hostname	Имя узла сети агента.	
agent.ping	Проверка доступности агента.	
agent.variant	Вариант агента Zabbix (агент Zabbix или агент Zabbix 2).	
agent.version	Версия агента Zabbix.	
zabbix.stats	Возвращает набор внутренних метрик сервера или прокси Zabbix. Если указаны `ip` и `port`, метрики извлекаются из удаленного экземпляра; в противном случае — из локального экземпляра.	
zabbix.stats (queue)	Возвращает количество отслеживаемых элементов данных в очереди, обработка которых задерживается на сервере или прокси Zabbix. Если указаны `ip` и `port`, метрики извлекаются из удаленного экземпляра; в противном случае — из локального экземпляра.	

Поддерживаемые платформы

Если в описании элемента данных не указано иное, элементы данных агента (и все параметры) поддерживаются на:

- **Linux**

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

Многие элементы данных агента также поддерживаются на **Windows**. Подробности смотрите на странице Элементы данных агента Windows.

Подробности ключей элементов данных

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками **< >**, опциональны.

kernel.maxfiles

 Максимальное количество открытых файлов, поддерживаемое ОС.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 Максимальное количество процессов, поддерживаемое ОС.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux 2.6 и новее, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 Количество открытых в данный момент файловых дескрипторов.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux (элемент данных может работать и на других UNIX-подобных платформах).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Мониторинг файла журнала.
 Возвращаемое значение: *Log*.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь и имя файла журнала;
- **regexp** - регулярное выражение, описывающее требуемый шаблон;
- **encoding** - идентификатор кодовой страницы;
- **maxlines** - максимальное число новых строк в секунду, которое агент будет отправлять на сервер Zabbix или прокси. Этот параметр переопределяет значение `MaxLinesPerSecond` в `zabbix_agentd.conf`;
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропуск обработки более старых данных (влияет только на вновь созданные элементы данных);
- **output** - необязательный шаблон форматирования вывода. Последовательность экранирования **\0** заменяется совпавшей частью текста (от первого символа, где начинается совпадение, до символа, где совпадение заканчивается), а последовательность **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает число захваченных групп);
- **maxdelay** - максимальная задержка в секундах. Тип: float. Значения:
  - 0 - (по умолчанию) никогда не игнорировать строки файла журнала;
  - > 0.0 - игнорировать более старые строки, чтобы наиболее свежие проанализированные строки были получены в пределах ”maxdelay” секунд. Перед использованием прочитайте примечания по maxdelay!
- **options** - дополнительные параметры:
mtime-noreread - неуникальные записи, перечитывать только если изменился размер файла (изменение времени модификации игнорируется). (Этот параметр устарел начиная с 5.0.2, так как теперь mtime игнорируется.)
- **persistent dir** (только в zabbix_agentd в системах Unix; не поддерживается в Zabbix agent 2) - абсолютный путь к каталогу, где хранятся постоянные файлы. См. также дополнительные примечания о постоянных файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Если файл отсутствует или права доступа не позволяют открыть его, элемент данных переходит в состояние неподдерживаемого.
- Если `output` оставить пустым, возвращается вся строка, содержащая совпавший текст. Обратите внимание, что все глобальные типы регулярных выражений, кроме *Matches regular expression*, всегда возвращают всю совпавшую строку, и параметр `output` игнорируется.
- Извлечение содержимого с использованием параметра `output` выполняется на агенте.

Примеры:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Пример использования параметра output для извлечения числа из записи журнала:

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

Пример использования параметра output для преобразования записи журнала перед отправкой на сервер:

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

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 Количество совпавших строк в контролируемом файле журнала.
 Возвращаемое значение: *Integer*.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь и имя файла журнала;
- **regexp** - регулярное выражение, описывающее требуемый шаблон;
- **encoding** - идентификатор кодовой страницы;
- **maxproclines** - максимальное количество новых строк в секунду, которые агент будет анализировать (не может превышать 10000). Значение по умолчанию: 10*MaxLinesPerSecond в zabbix_agentd.conf.
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропускать обработку более старых данных (влияет только на вновь созданные элементы данных).
- **maxdelay** - максимальная задержка в секундах. Тип: float. Значения:
  - 0 - (по умолчанию) никогда не игнорировать строки файла журнала;
  - > 0.0 - игнорировать более старые строки, чтобы наиболее свежие строки были проанализированы в течение ”maxdelay” секунд. Перед использованием прочитайте примечания по maxdelay!
- **options** - дополнительные параметры:
mtime-noreread - неуникальные записи, перечитывать только если изменяется размер файла (игнорировать изменение времени модификации). (Этот параметр устарел начиная с версии 5.0.2, поскольку теперь mtime игнорируется.)
- **persistent dir** (только в zabbix_agentd в системах Unix; не поддерживается в Zabbix agent 2) - абсолютный путь к каталогу, где хранятся постоянные файлы. См. также дополнительные примечания о постоянных файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Совпадающие строки подсчитываются среди новых строк с момента последней проверки журнала агентом, и поэтому зависят от интервала обновления элемента данных.
- Если файл отсутствует или доступ к нему запрещен из-за прав доступа, элемент данных переходит в состояние неподдерживаемого.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Мониторинг ротируемого файла журнала.
 Возвращаемое значение: *Log*.
 См. поддерживаемые платформы.

Параметры:

- **file regexp** - абсолютный путь к файлу, при этом имя файла задается с использованием регулярного выражения. Обратите внимание, что регулярное выражение применяется только к имени файла и не должно совпадать с полным именем (например, /path/to/agent будет соответствовать zabbix_agentd.log).
- **regexp** - регулярное выражение, описывающее требуемый шаблон содержимого.
- **encoding** - идентификатор кодовой страницы.
- **maxlines** - максимальное число новых строк в секунду, которое агент будет отправлять на сервер Zabbix или прокси. Этот параметр переопределяет значение MaxLinesPerSecond в zabbix_agentd.conf.
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропуск обработки более старых данных (влияет только на вновь созданные элементы данных).
- **output** - необязательный шаблон форматирования вывода. Последовательность экранирования **\0** заменяется совпавшей частью текста (от первого символа, где начинается совпадение, до символа, где совпадение заканчивается), а последовательность **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает число захваченных групп).
- **maxdelay** - максимальная задержка в секундах. Тип: float. Значения:
  - 0 - (по умолчанию) никогда не игнорировать строки файла журнала;
  - > 0.0 - игнорировать более старые строки, чтобы наиболее свежие строки были проанализированы в течение ”maxdelay” секунд. Перед использованием прочитайте примечания о maxdelay!

- **options** - тип ротации файла журнала и другие параметры. Возможные значения:
  - *rotate* (по умолчанию).
  - *copytruncate* - обратите внимание, что *copytruncate* нельзя использовать вместе с *maxdelay*. В этом случае *maxdelay* должен быть 0 или не указан; см. примечания о copytruncate.
  - *mtime-reread* - неуникальные записи, перечитывать при изменении времени модификации или размера (используется по умолчанию, если явно не задан ни один параметр `mtime-*`).
  - *mtime-noreread* - неуникальные записи, перечитывать только при изменении размера (игнорировать изменение времени модификации).
- **persistent dir** (только в `zabbix_agentd` в системах Unix; не поддерживается в Zabbix agent 2) - абсолютный путь к каталогу, где будут храниться постоянные файлы. См. также дополнительные примечания о постоянных файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Ротация журнала основана на времени последнего изменения файлов.
- Обратите внимание, что `logrt` предназначен для работы с одним текущим активным файлом журнала и несколькими другими совпадающими неактивными ротированными файлами. Если, например, в каталоге много активных файлов журнала, для каждого из них следует создать отдельный элемент данных `logrt`. В противном случае, если один элемент данных `logrt` будет обрабатывать слишком много файлов, это может привести к исчерпанию памяти и сбою мониторинга.
- Если `output` оставить пустым, будет возвращена вся строка, содержащая совпавший текст. Обратите внимание, что все глобальные типы регулярных выражений, кроме *Matches regular expression*, всегда возвращают всю совпавшую строку, а параметр `output` игнорируется.
- Извлечение содержимого с использованием параметра `output` выполняется на агенте.
- В параметре `file regexp` путь к каталогу журнала и регулярное выражение имени файла журнала должны быть разделены правильным разделителем каталогов:
  - В Windows разделителем должен быть обратный слеш (\); прямые слеши могут допускаться в других позициях, кроме той, которая разделяет путь к каталогу журнала и регулярное выражение имени файла журнала (см. примеры ниже).
  - В других системах должен использоваться прямой слеш (/).

Примеры для Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #                                  "c:/d
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #
```

Примеры для других систем:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #                                  "logfile1" (
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #
```

Пример использования параметра `output` для извлечения числа из записи журнала:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #
```

Пример использования параметра `output` для преобразования записи журнала перед отправкой на сервер:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 Количество совпавших строк в контролируемом журнале, который подвергается ротации.
 Возвращаемое значение: *Integer*.
 См. поддерживаемые платформы.

Параметры:

- **file regexp** - абсолютный путь к файлу, при этом имя файла задается с помощью регулярного выражения. Обратите внимание, что регулярное выражение применяется только к имени файла и не должно совпадать с полным именем целиком (например, `/path/to/agent` будет соответствовать `zabbix_agentd.log`).
- **regexp** - регулярное выражение, описывающее требуемый шаблон.
- **encoding** - идентификатор кодовой страницы identifier.
- **maxproclines** - максимальное количество новых строк в секунду, которое агент будет анализировать (не может превышать 10000). Значение по умолчанию: 10*MaxLinesPerSecond в `zabbix_agentd.conf`.
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропуск обработки более старых данных (влияет только на вновь созданные элементы данных).
- **maxdelay** - максимальная задержка в секундах. Тип: float. Значения:
  - 0 - (по умолчанию) никогда не игнорировать строки журнала;

    - > 0.0 - игнорировать более старые строки, чтобы наиболее свежие строки были проанализированы в течение "maxdelay" секунд. Перед использованием прочитайте примечания к maxdelay!
- **options** - тип ротации файла журнала и другие параметры. Возможные значения:
    - *rotate* (по умолчанию).
    - *copytruncate* - обратите внимание, что copytruncate нельзя использовать вместе с maxdelay. В этом случае maxdelay должен быть 0 или не указан; см. примечания к copytruncate.
    - *mtime-reread* - неуникальные записи, перечитывать при изменении времени модификации или размера (по умолчанию).
    - *mtime-noreread* - неуникальные записи, перечитывать только при изменении размера (игнорировать изменение времени модификации).
- **persistent dir** (только в zabbix_agentd в системах Unix; не поддерживается в Zabbix agent 2) - абсолютный путь к каталогу, где будут храниться постоянные файлы. См. также дополнительные примечания о постоянных файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Совпавшие строки подсчитываются среди новых строк с момента последней проверки журнала агентом и, таким образом, зависят от интервала обновления элемента данных.
- Ротация журнала основана на времени последнего изменения файлов.
- В параметре file regexp путь к каталогу журнала и регулярное выражение файла журнала должны быть разделены правильным разделителем каталогов:
    - В Windows разделителем должен быть обратный слеш (\); прямые слеши могут допускаться в других позициях, кроме той, которая разделяет путь к каталогу журнала и регулярное выражение файла журнала (см. примеры ниже).
    - В других системах он должен быть прямым слешем (/).

Примеры для Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will count the number of ma
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will count
```

Примеры для других систем:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #this item will count the number of matches
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #this item will count the
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Читает данные Modbus.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **endpoint** - endpoint, определенный как protocol://connection_string;
- **slave id** - ID slave;
- **function** - функция Modbus;
- **address** - адрес первого регистра, coil или input;
- **count** - количество записей для чтения;
- **type** - тип данных;
- **endianness** - настройка порядка байтов;
- **offset** - количество регистров, начиная с address, результаты которых будут отброшены.

В Windows этот элемент данных следует по UNC-путям в параметре endpoint, что может представлять риск безопасности.

См. подробное описание параметров.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Проверяет состояние службы DNS.
 Возвращаемые значения: - 0 - разрешение DNS не удалось (DNS-сервер не ответил или вернул ошибку); - 1 - разрешение DNS выполнено успешно.
 См. поддерживаемые платформы.

Параметры:

- **ip** (игнорируется в Windows, если не используется Zabbix agent 2) - IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** - DNS-имя для запроса;
- **type** - тип записи, которую нужно запросить (по умолчанию SOA);
- **timeout** (игнорируется в Windows, если не используется Zabbix agent 2) - время ожидания запроса в секундах (по умолчанию 1 секунда);

- **count** (игнорируется в Windows, если не используется Zabbix agent 2) - количество попыток выполнения запроса (по умолчанию 2);
- **protocol** - протокол, используемый для выполнения DNS-запросов: *udp* (по умолчанию) или *tcp*.

Комментарии:

- Возможные значения `type`: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (не поддерживается для Zabbix agent в Windows, Zabbix agent 2 во всех ОС), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Для обратного DNS-поиска (когда `type` установлен в *PTR*) можно указывать DNS-имя как в обратном, так и в прямом формате (см. примеры ниже). Обратите внимание, что при запросе записи PTR DNS-имя фактически является IP-адресом.
- Имена доменов с международными символами не поддерживаются, используйте вместо них имена, закодированные в IDNA.

Примеры:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Проверяет производительность службы DNS.
 Возвращаемое значение: *Float* (0 - служба недоступна; секунды - количество секунд, затраченных на ожидание ответа от службы).
 См. поддерживаемые платформы.

Параметры:

- **ip** (игнорируется в Windows, если не используется Zabbix agent 2) - IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** - DNS-имя для запроса;
- **type** - тип записи, которую нужно запросить (по умолчанию `SOA`);
- **timeout** (игнорируется в Windows, если не используется Zabbix agent 2) - время ожидания запроса в секундах (по умолчанию 1 секунда);
- **count** (игнорируется в Windows, если не используется Zabbix agent 2) - количество попыток для запроса (по умолчанию 2);
- **protocol** - протокол, используемый для выполнения DNS-запросов: *udp* (по умолчанию) или *tcp*.

Комментарии:

- Возможные значения для `type`:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (не поддерживается для Zabbix agent в Windows, Zabbix agent 2 во всех ОС), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Для обратных DNS-запросов (когда `type` установлен в *PTR*) можно указать DNS-имя как в обратном, так и в прямом формате (см. примеры ниже). Обратите внимание, что при запросе записи PTR DNS-имя фактически является IP-адресом.
- Имена доменов с интернационализацией не поддерживаются, используйте вместо них имена, закодированные в IDNA.
- Элемент данных возвращает время ответа вместо 0, если DNS-сервер отвечает кодом ошибки (например, `NXDOMAIN` или `SERVFAIL`).

Примеры:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Выполняет DNS-запрос.

Агент Zabbix 2 также предоставляет net.dns.get, который предлагает дополнительные возможности, такие как больше типов записей и более гибкое управление мониторингом DNS. Возвращаемое значение: строка символов с информацией требуемого типа.
 См. поддерживаемые платформы.

Параметры:

- **ip** (игнорируется в Windows, если не используется агент Zabbix 2) - IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** - DNS-имя для запроса;
- **type** - тип записи, которую нужно запросить (по умолчанию SOA);
- **timeout** (игнорируется в Windows, если не используется агент Zabbix 2) - время ожидания запроса в секундах (по умолчанию 1 секунда);
- **count** (игнорируется в Windows, если не используется агент Zabbix 2) - количество попыток для запроса (по умолчанию 2);
- **protocol** - протокол, используемый для выполнения DNS-запросов: *udp* (по умолчанию) или *tcp*.

Комментарии:

- Возможные значения type:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (не поддерживается для агента Zabbix в Windows, агент Zabbix 2 на всех ОС), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Для обратных DNS-запросов (когда type установлен в *PTR*) можно указать DNS-имя в перевернутом или обычном формате (см. примеры ниже). Обратите внимание, что при запросе записи PTR DNS-имя фактически является IP-адресом.
- Имена доменов с международными символами не поддерживаются, используйте вместо них имена, закодированные в IDNA.

Примеры:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 Количество коллизий за пределами окна (out-of-window).
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. На NetBSD требуются права суперпользователя (root).

Параметр:

- **if** — имя сетевого интерфейса

net.if.discovery

 Список сетевых интерфейсов. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.get[<regexp>]

 Список сетевых интерфейсов с подробной информацией.
 Используется для низкоуровневого обнаружения.
 Этот элемент данных является расширенной версией элемента данных `net.if.discovery` агента Zabbix.
 Возвращаемое значение: *JSON object*.
 Поддерживаемые платформы: Linux.

Параметры:

- **regexp** - регулярное выражение, соответствующее именам сетевых интерфейсов.

Комментарии:

- Возвращает JSON-объект с сетевыми интерфейсами, имена которых соответствуют заданному регулярному выражению (см. подробнее).

net.if.in[if,<mode>]

 Статистика входящего трафика на сетевом интерфейсе.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Для NetBSD требуются права root.

Параметры:

- **if** - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4-адрес; или, если указано в фигурных скобках, GUID сетевого интерфейса (Windows);
- **mode** - возможные значения:
  - *bytes* - количество байтов (по умолчанию);

  - *packets* - количество пакетов;
  - *errors* - количество ошибок;
  - *dropped* - количество отброшенных пакетов;
  - *overruns (fifo)* - количество ошибок буфера FIFO;
  - *frame* - количество ошибок кадрирования пакетов;
  - *compressed* - количество сжатых пакетов, полученных драйвером устройства;
  - *multicast* - количество многоадресных кадров, полученных драйвером устройства.

Комментарии:

- Вы можете использовать этот ключ с шагом предварительной обработки *Change per second*, чтобы получить статистику в байтах в секунду.
- Режим *dropped* поддерживается только в Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Режимы *overruns*, *frame*, *compressed*, *multicast* поддерживаются только в Linux.
- В HP-UX этот элемент данных не предоставляет подробные сведения для интерфейсов loopback (например, lo0).

Примеры:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Статистика исходящего трафика на сетевом интерфейсе.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. На NetBSD требуются права root.

Параметры:

- **if** - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4-адрес; либо, если указано в фигурных скобках, GUID сетевого интерфейса (Windows);
- **mode** - возможные значения:
  - *bytes* - количество байт (по умолчанию);
  - *packets* - количество пакетов;
  - *errors* - количество ошибок;
  - *dropped* - количество отброшенных пакетов;
  - *overruns (fifo)* - количество ошибок буфера FIFO;
  - *collisions (colls)* - количество коллизий, обнаруженных на интерфейсе;
  - *carrier* - количество потерь carrier, обнаруженных драйвером устройства;
  - *compressed* - количество сжатых пакетов, переданных драйвером устройства.

Комментарии:

- Вы можете использовать этот ключ с шагом предварительной обработки *Change per second*, чтобы получить статистику в байтах в секунду.
- Режим *dropped* поддерживается только в Linux, HP-UX.
- Режимы *overruns*, *collision*, *carrier*, *compressed* поддерживаются только в Linux.
- В HP-UX этот элемент данных не предоставляет подробные сведения о loopback-интерфейсах (например, lo0).

Примеры:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 Сумма статистики входящего и исходящего трафика на сетевом интерфейсе.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. На NetBSD требуются права root.

Параметры:

- **if** - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4-адрес; или, если указано в фигурных скобках, GUID сетевого интерфейса (Windows);
- **mode** - возможные значения:
  - *bytes* - количество байт (по умолчанию);
  - *packets* - количество пакетов;
  - *errors* - количество ошибок;
  - *dropped* - количество отброшенных пакетов;
  - *overruns (fifo)* - количество ошибок буфера FIFO;
  - *collisions (colls)* - количество коллизий, обнаруженных на интерфейсе;
  - *compressed* - количество сжатых пакетов, переданных или полученных драйвером устройства.

Комментарии:

- Вы можете использовать этот ключ с шагом предварительной обработки *Change per second*, чтобы получить статистику в байтах в секунду.
- Режим *dropped* поддерживается только в Linux, HP-UX. Отброшенные пакеты поддерживаются только в том случае, если и `net.if.in`, и `net.if.out` работают для отброшенных пакетов на вашей платформе.
- Режимы *overruns*, *collision*, *compressed* поддерживаются только в Linux.
- В HP-UX этот элемент данных не предоставляет подробности для loopback-интерфейсов (например, lo0).

Примеры:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Проверяет, находится ли этот TCP-порт в состоянии LISTEN.
 Возвращаемые значения: 0 - не находится в состоянии LISTEN; 1 - находится в состоянии LISTEN.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, MacOS X, Windows.

Параметр:

- **port** - номер TCP-порта

В ядрах Linux 2.6.14 и выше информация о прослушивающих TCP-сокетах, если возможно, получается через интерфейс NETLINK ядра. В противном случае информация извлекается из файлов `/proc/net/tcp` и `/roc/net/tcp6`.

Пример:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Проверяет, возможно ли установить TCP-соединение с указанным портом.
 Возвращаемые значения: 0 - подключиться нельзя; 1 - подключиться можно.
 См. поддерживаемые платформы.

Параметры:

- **ip** - IP-адрес или DNS-имя (по умолчанию `127.0.0.1`);
- **port** - номер порта.

Комментарии:

- Для простого тестирования производительности TCP используйте `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Эти проверки могут приводить к появлению дополнительных сообщений в журналах системных демонов (обычно в журналах сеансов SMTP и SSH).

Пример:

```
net.tcp.port[,80] #                                  -    ,              80
```

net.tcp.service[service,<ip>,<port>]

 Проверяет, запущена ли служба и принимает ли она TCP-соединения.
 Возвращаемые значения: 0 - служба недоступна; 1 - служба запущена.
 См. поддерживаемые платформы.

Параметры:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* или *telnet* (см. подробности);
- **ip** - IP-адрес или DNS-имя (по умолчанию `127.0.0.1`);
- **port** - номер порта (по умолчанию используется стандартный номер порта службы).

Комментарии:

- Эти проверки могут приводить к появлению дополнительных сообщений в системных журналах демонов (обычно регистрируются сеансы SMTP и SSH).
- Проверка зашифрованных протоколов (например, IMAP на порту 993 или POP на порту 995) в настоящее время не поддерживается. В качестве обходного решения используйте `net.tcp.port[]` для подобных проверок.
- Проверка LDAP и HTTPS в Windows поддерживается только агентом Zabbix agent 2.
- Проверка telnet ищет приглашение к входу в систему (':' в конце).

Пример:

```
net.tcp.service[ftp,,45] #this item can be used to test the availability of FTP server on TCP port 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Проверяет производительность TCP-сервиса.
 Возвращаемые значения: *Float* (0 - сервис недоступен; seconds - количество секунд, затраченных на ожидание ответа от сервиса).
 См. поддерживаемые платформы.

Параметры:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* или *telnet* (см. подробности);
- **ip** - IP-адрес или DNS-имя (по умолчанию 127.0.0.1);
- **port** - номер порта (по умолчанию используется стандартный номер порта сервиса).

Комментарии:

- Проверка зашифрованных протоколов (например, IMAP на порту 993 или POP на порту 995) в настоящее время не поддерживается. В качестве обходного решения используйте `net.tcp.service.perf[tcp,<ip>,<port>]` для таких проверок.
- Проверка telnet ищет приглашение к входу в систему (':' в конце).

Пример:

```
net.tcp.service.perf[ssh] #                                                SSH-
```

net.tcp.socket.count[<локальный_адрес>,<локальный_порт>,<удалённый_адрес>,<удалённый_порт>,<состояние>]

 Получение количества TCP сокетов, соответствующих заданным параметрам.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **локальный_адрес** — локальный IPv4/6 адрес или CIDR подсеть;
- **локальный_порт** — номер локального порта или имя сервиса;
- **удалённый_адрес** — удалённый IPv4/6 адрес или CIDR подсеть;
- **удалённый_порт** — номер удалённого порта или имя сервиса;
- **состояние** — состояние соединения (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Пример:

```
net.tcp.socket.count[,80,,,established] #                  TCP     80         «established»
```

net.udp.listen[порт]

 Проверка, находится ли UDP порт в состоянии LISTEN.
 Возвращаемые значения: 0 — не находится в состоянии LISTEN; 1 — находится в состоянии LISTEN.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, MacOS X.

Параметр:

- **порт** — номер UDP порта

Пример:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Проверяет, запущена ли служба и отвечает ли она на UDP-запросы.
 Возвращаемые значения: 0 - служба недоступна; 1 - служба запущена.
 См. поддерживаемые платформы.

Параметры:

- **service** - *ntp* (см. подробности);
- **ip** - IP-адрес или DNS-имя (по умолчанию 127.0.0.1);
- **port** - номер порта (по умолчанию используется стандартный номер порта службы).

Пример:

```
net.udp.service[ntp,,45] #                                          NTP    UDP-     45
```

net.udp.service.perf[service,<ip>,<port>]

 Проверяет производительность UDP-сервиса.
 Возвращаемые значения: *Float* (0 - сервис недоступен; seconds - количество секунд, затраченных на ожидание ответа от сервиса).
 См. поддерживаемые платформы.

Параметры:

- **service** - *ntp* (см. подробности);

- **ip** - IP-адрес или DNS-имя (по умолчанию 127.0.0.1);
- **port** - номер порта (по умолчанию используется стандартный номер порта сервиса).

Пример:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<локальный_адрес>,<локальный_порт>,<удалённый_адрес>,<удалённый_порт>,<состояние>]

 Получение количества UDP сокетов, соответствующих заданным параметрам.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **локальный_адрес** — локальный IPv4/6 адрес или CIDR подсеть;
- **локальный_порт** — номер локального порта или имя сервиса;
- **удалённый_адрес** — удалённый IPv4/6 адрес или CIDR подсеть;
- **удалённый_порт** — номер удалённого порта или имя сервиса;
- **состояние** — состояние соединения (*established*, *unconn*).

Пример:

```
net.udp.socket.count[,,,,established] #            UDP            «established»
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 Процент использования CPU процессом.
 Возвращаемое значение: *Float*.
 Поддерживаемые платформы: Linux, Solaris[6].

Параметры:

- **name** - имя процесса (по умолчанию *all processes*);
- **user** - имя пользователя (по умолчанию *all users*);
- **type** - тип использования CPU: *total* (по умолчанию), *user* или *system*;
- **cmdline** - фильтрация по командной строке (это регулярное выражение);
- **mode** - режим сбора данных: *avg1* (по умолчанию), *avg5* или *avg15*;
- **zone** - целевая зона: *current* (по умолчанию) или *all*. Этот параметр поддерживается только в Solaris.

Комментарии:

- Возвращаемое значение основано на проценте использования одного ядра CPU. Например, использование CPU процессом, полностью загружающим два ядра, составляет 200%.
- Данные об использовании CPU процессом собираются сборщиком, который поддерживает максимум 1024 уникальных запросов (по имени, пользователю и командной строке). Запросы, к которым не обращались в течение последних 24 часов, удаляются из сборщика.
- При установке параметра zone в значение *current* (или по умолчанию), если агент был скомпилирован в Solaris без поддержки зон, но запущен в более новой версии Solaris, где зоны поддерживаются, агент вернет NOTSUPPORTED (агент не может ограничить результаты только текущей зоной). Однако в этом случае поддерживается значение *all*.

Примеры:

```
proc.cpu.util[,root] #        CPU          ,                "root"
proc.cpu.util[zabbix_server,zabbix] #        CPU          zabbix_server,                zabbix
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 Список процессов ОС и их параметров. Может использоваться для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Параметры:

- **name** - имя процесса (по умолчанию *all processes*).
- **user** - имя пользователя (по умолчанию *all users*).
- **cmdline** - фильтрация по командной строке (это регулярное выражение). Этот параметр не поддерживается в Windows; на других платформах он не поддерживается, если для mode задано значение 'summary'.
- **mode** - возможные значения:
  - *process* (по умолчанию);
  - *thread* (не поддерживается в NetBSD);
  - *summary*. См. список параметров процесса, возвращаемых для каждого режима и ОС.

Комментарии:

- Если значение невозможно получить, например из-за ошибки (процесс уже завершился, недостаточно прав, сбой системного вызова), будет возвращено -1.
- См. примечания по выбору процессов с параметрами name и cmdline (только для Linux).

Примеры:

```
proc.get[zabbix_server,zabbix,,process] #             zabbix_server,                zabbix,
proc.get[java,,,thread] #             Java,
proc.get[,zabbix,,summary] #                         ,                zabbix,
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 Память, используемая процессом, в байтах.
 Возвращаемое значение: - *Integer* - при *mode* = *max*, *min*, *sum*; - *Float* - при *mode* = *avg*
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Параметры:

- **name** - имя процесса (по умолчанию - *all processes*);
- **user** - имя пользователя (по умолчанию - *all users*);
- **mode** - возможные значения: *avg*, *max*, *min* или *sum* (по умолчанию);
- **cmdline** - фильтрация по командной строке (это регулярное выражение);
- **memtype** - тип памяти, используемой процессом

Комментарии:

- Параметр memtype поддерживается только в Linux, FreeBSD, Solaris[6], AIX.
- Если несколько процессов используют общую память, сумма памяти, используемой процессами, может привести к большим нереалистичным значениям.

См. примечания по выбору процессов с параметрами name и cmdline (только для Linux).

Когда этот элемент данных вызывается из командной строки и содержит параметр командной строки (например, при использовании тестового режима агента: zabbix_agentd -t proc.mem[,,,apache2]), будет учтен один дополнительный процесс, так как агент будет считать и себя.

Примеры:

```
proc.mem[,root] #    ,                  ,                  "root"
proc.mem[zabbix_server,zabbix] #    ,                  zabbix_server,                  zabbix
proc.mem[,oracle,max,oracleZABBIX] #    ,              "      "             ,          Oracle          oracleZ
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Количество процессов.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Параметры:

- **name** - имя процесса (по умолчанию - *all processes*).
- **user** - имя пользователя (по умолчанию - *all users*).
- **state** - возможные значения:
  - *all* (по умолчанию);
  - *disk* - непрерываемый сон;
  - *run* - выполняется;
  - *sleep* - прерываемый сон;
  - *trace* - остановлен;
  - *zomb* - зомби.
- **cmdline** - фильтрация по командной строке (это регулярное выражение).
- **zone** - целевая зона: *current* (по умолчанию) или *all*. Этот параметр поддерживается только в Solaris.

Комментарии:

- Параметры состояния *disk* и *trace* поддерживаются только в Linux, FreeBSD, OpenBSD, NetBSD.
- В Windows поддерживаются только параметры name и user.
- Когда этот элемент данных вызывается из командной строки и содержит параметр командной строки (например, при использовании тестового режима агента: zabbix_agentd -t proc.num[,,,apache2]), будет подсчитан еще один процесс, так как агент посчитает и себя.
- При установке параметра zone в значение current (или по умолчанию), если агент был скомпилирован в Solaris без поддержки зон, но запущен в более новой версии Solaris, где зоны поддерживаются, агент вернет NOTSUPPORTED (агент не может ограничить результаты только текущей зоной). Однако в этом случае поддерживается значение *all*.
- См. примечания по выбору процессов с параметрами name и cmdline (только для Linux).

Примеры:

```
proc.num[,mysql] #           ,                mysql
proc.num[apache2,www-data] #            apache2,                www-data
proc.num[,oracle,sleep,oracleZABBIX] #                    sleep,          Oracle          oracleZABBIX
```

sensor[device,sensor,<mode>]

 Показания аппаратного датчика.
 Возвращаемое значение: *Float*.
 Поддерживаемые платформы: Linux, OpenBSD.

Параметры:

- **device** - имя устройства;
- **sensor** - имя датчика;
- **mode** - возможные значения: *avg*, *max* или *min*.

Комментарии:

- Читает `/proc/sys/dev/sensors` в Linux 2.4.
- Читает `/sys/class/hwmon` в Linux 2.6+. См. более подробное описание элемента данных sensor в Linux.
- Читает MIB `hw.sensors` в OpenBSD.
- В Linux 2.4, если указан mode, то device и sensor рассматриваются как регулярные выражения.
- В Linux 2.6+ и OpenBSD, либо если mode опущен, device и sensor рассматриваются как буквальные значения.

Примеры:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #           CPU
sensor["cpu[0-2]$",temp,avg] #                    CPU
```

system.boottime

 Штамп времени (timestamp) загрузки системы.
 Возвращаемое значение: *Целое число (Unix timestamp)*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 Список обнаруженных CPU/ядер CPU. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON string*.
 См. supported platforms.

system.cpu.intr

 Прерывания устройств.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<режим>]

 Загрузка CPU.
 Возвращаемое значение: *Число с плавающей точкой*.
 Смотрите Поддерживаемые платформы.

Параметры:

- **cpu** — возможные значения: *all* (по умолчанию) или *percpu* (общая загрузка делится на количество CPU онлайн);
- **режим** — возможные значения: *avg1* (усреднение за одну минуту, по умолчанию), *avg5* или *avg15*.

Пример:

```
system.cpu.load[,avg5]
```

system.cpu.num[<тип>]

 Количество CPU.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Параметр:

- **тип** — возможные значения: *online* (по умолчанию) или *max*

Значение *max* параметра «тип» поддерживается только в Linux, FreeBSD, Solaris, MacOS X.

Пример:

```
system.cpu.num
```

system.cpu.switches

 Количество переключений контекста.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 Процент использования CPU.
 Возвращаемое значение: *Float*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Параметры:

- **cpu** - *<номер CPU>* или *all* (по умолчанию);
- **type** - возможные значения: *user* (по умолчанию), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (в OpenBSD), *guest* (в ядрах Linux 2.6.24 и выше) или *guest_nice* (в ядрах Linux 2.6.33 и выше);
- **mode** - возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*;
- **logical or physical** - возможные значения: *logical* (по умолчанию) или *physical*. Этот параметр поддерживается только в AIX.

Комментарии:

- Параметр типа *nice* поддерживается только в Linux, FreeBSD, HP-UX, OpenBSD, NetBSD.
- Параметр типа *iowait* поддерживается только в Linux 2.6 и новее, Solaris, AIX.
- Параметр типа *interrupt* поддерживается только в Linux 2.6 и новее, FreeBSD, OpenBSD.
- Параметры типа *softirq*, *steal*, *guest*, *guest_nice* поддерживаются только в Linux 2.6 и новее.
- Параметры режима *avg5* и *avg15* поддерживаются в Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

Пример:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 Имя узла системы.
 Возвращаемое значение: *String*.
 См. поддерживаемые платформы.

Параметры:

- **type** - возможные значения:
  - *netbios* (по умолчанию в Windows);
  - *host* (по умолчанию в Linux);
  - *shorthost* (возвращает часть имени узла до первой точки, полную строку для имен без точек);
  - *fqdn* (возвращает Fully Qualified Domain Name).
- **transform** - возможные значения:
  - *none* (по умолчанию);
  - *lower* (преобразовать в нижний регистр).

Значение получается путем взятия `nodename` из вывода системного API uname().

Примеры возвращаемых значений:

```
system.hostname → linux-w7x1
system.hostname → example.com
system.hostname[shorthost] → example
system.hostname → WIN-SERV2008-I6
system.hostname[host] → Win-Serv2008-I6LonG
system.hostname[host,lower] → win-serv2008-i6long
system.hostname[fqdn,lower] → blog.zabbix.com
```

system.hw.chassis[<информация>]

 Информация о шасси.
 Возвращаемое значение: *Строка*.
 Поддерживаемые платформы: Linux.

Параметр:

- **информация** — возможные значения: *full* (по умолчанию), *model*, *serial*, *type* или *vendor*

Комментарии:

- Этот ключ элемента данных зависит от доступности таблицы SMBIOS;
- Будет выполнена попытка чтения таблицы DMI из sysfs; если доступ к sysfs будет неудачным, тогда будет попытка чтения напрямую из памяти;
- Требуются **права root**, потому что значение читается из sysfs или памяти.

Пример:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<информация>]

 Информация о CPU.
 Возвращаемое значение: *Строка* или *Целое число*.
 Поддерживаемые платформы: Linux.

Параметры:

- **cpu** — <*номер CPU*> или *all* (по умолчанию);
- **информация** — возможные значения: *full* (по умолчанию), *curfreq*, *maxfreq*, *model* или *vendor*.

Комментарии:

- Информация собирается из `/proc/cpuinfo` и из `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`;
- Если указаны номер CPU и *curfreq* или *maxfreq*, то возвращается числовое значение (Гц).

Пример:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<тип>]

 Список PCI или USB устройств.
 Возвращаемое значение: *Текст*.
 Поддерживаемые платформы: Linux.

Параметр:

- **тип** — *pci* (по умолчанию) или *usb*

Возвращает вывод утилит lspci или lsusb (выполняются без каких либо параметров).

Пример:

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

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

 Список MAC-адресов.
 Возвращаемое значение: *String*.
 Поддерживаемые платформы: Linux.

Параметры:

- **interface** - *all* (по умолчанию) или регулярное выражение;
- **format** - *full* (по умолчанию) или *short*.

Комментарии:

- Выводит MAC-адреса интерфейсов, имя которых соответствует заданному регулярному выражению `interface` (*all* выводит список для всех интерфейсов).
- Если для `format` указано значение *short*, имена интерфейсов и одинаковые MAC-адреса не выводятся.

Пример:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 Системное время.
 Возвращаемое значение: *Integer* - при `type` = *utc*; *String* - при `type` = *local*.
 См. поддерживаемые платформы.

Параметры:

- **type** - возможные значения:
  - *utc* - (по умолчанию) время, прошедшее с начала эпохи (00:00:00 UTC, 1 января 1970 года), измеряемое в секундах;
  - *local* - время в формате 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Должен использоваться как пассивная проверка для агента Zabbix; в Zabbix agent 2 может быть настроен как активная проверка.

Пример:

```
system.localtime[local] #create an item using this key and then use it to display the host time in the *Cl
```

system.run[command,<mode>]

 Выполняет указанную команду на узле сети.
 Возвращаемое значение: результат команды типа *Text* или 1 - при `mode` = *nowait* (независимо от результата команды).
 См. поддерживаемые платформы.

Параметры:

- **command** - команда для выполнения;
- **mode** - возможные значения:
  - *wait* - ждать завершения выполнения (по умолчанию);
  - *nowait* - не ждать.

Комментарии:

- Этот элемент данных по умолчанию отключен. Узнайте, как включить их.
- Возвращаемое значение элемента данных - это стандартный вывод вместе со стандартным потоком ошибок, созданным командой. Проверка кода завершения не выполняется.
- Для корректной обработки возвращаемое значение команды должно иметь тип данных `text`. Пустой результат также допускается.
- Возвращаемое значение ограничено 16 МБ (включая завершающие пробелы, которые обрезаются); также действуют ограничения базы данных.
- См. также: Выполнение команд.

Пример:

```
system.run[ls -l /] #return a detailed file list of the root directory
```

system.stat[resource,<type>]

 Системная статистика.
 Возвращаемое значение: *Integer* или *float*.
 Поддерживаемые платформы: AIX.

Параметры:

- **ent** - количество процессорных единиц, которое данному разделу разрешено получать (float).
- **kthr,<type>** - информация о состояниях потоков ядра:
  - *r* - среднее количество исполняемых потоков ядра (float);
  - *b* - среднее количество потоков ядра, помещенных в очередь ожидания Virtual Memory Manager (float).
- **memory,<type>** - информация об использовании виртуальной и физической памяти:
  - *avm* - активные виртуальные страницы (integer);
  - *fre* - размер списка свободных страниц (integer).
- **page,<type>** - информация о page fault и активности подкачки:
  - *fi* - количество загрузок страниц из файла в секунду (float);
  - *fo* - количество выгрузок страниц в файл в секунду (float);
  - *pi* - страницы, загруженные из пространства подкачки (float);
  - *po* - страницы, выгруженные в пространство подкачки (float);
  - *fr* - освобожденные страницы (замена страниц) (float);
  - *sr* - страницы, просмотренные алгоритмом замены страниц (float).
- **faults,<type>** - частота ловушек и прерываний:
  - *in* - прерывания устройств (float);
  - *sy* - системные вызовы (float);
  - *cs* - переключения контекста потоков ядра (float).
- **cpu,<type>** - распределение процентного использования процессорного времени:
  - *us* - пользовательское время (float);
  - *sy* - системное время (float);
  - *id* - время простоя (float);
  - *wa* - время простоя, в течение которого в системе были невыполненные запросы ввода-вывода на диск/NFS (float);
  - *pc* - количество задействованных физических процессоров (float);
  - *ec* - процент использованной выделенной емкости (float);
  - *lbusy* - указывает процент использования логических процессоров, который приходился на выполнение на пользовательском и системном уровнях (float);
  - *app* - указывает доступные физические процессоры в общем пуле (float).
- **disk,<type>** - статистика диска:
  - *bps* - указывает объем данных, переданных на накопитель или с него, в байтах в секунду (integer);
  - *tps* - указывает количество передач в секунду, выполненных для физического диска/ленты (float).

Комментарии:

- Обратите внимание на следующие ограничения для этих элементов данных:
  - `system.stat[cpu,app]` - поддерживается только на AIX LPAR типа ”Shared”;
  - `system.stat[cpu,ec]` - поддерживается на AIX LPAR типов ”Shared” и ”Dedicated” (”Dedicated” всегда возвращает 100 (percent));
  - `system.stat[cpu,lbusy]` - поддерживается только на AIX LPAR типа ”Shared”;

- system.stat[cpu,pc] - поддерживается на AIX LPAR типов "Shared" и "Dedicated";
- system.stat[ent] - поддерживается на AIX LPAR типов "Shared" и "Dedicated".

system.sw.arch

 Информация об архитектуре программного обеспечения.
 Возвращаемое значение: *Строка*.
 Смотрите Поддерживаемые платформы.

Для получения информации используется функция uname().

Пример:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Информация об операционной системе.
 Возвращаемое значение: *String*.
 Поддерживаемые платформы: Linux, Windows.

Параметр:

- **info** - возможные значения: *full* (по умолчанию), *short* или *name*

Информация info получается из следующих источников (обратите внимание, что не все файлы и параметры присутствуют во всех дистрибутивах):

- /proc/version (*full*) в Linux;
- /proc/version_signature (*short*) в Linux;
- параметр PRETTY_NAME из /etc/os-release в системах Linux, поддерживающих его, или /etc/issue.net (*name*);
- ключ реестра HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion в Windows.

Примеры:

```
system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11
system.sw.os[full] → [s|Windows 10 Enterprise 22621.1.asd64fre.ni_release.220506-1250 Build 22621.963]
```

system.sw.os.get

 Подробная информация об операционной системе (версия, тип, название дистрибутива, младшая и старшая версии и т. д.).
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 Список установленных пакетов.
 Возвращаемое значение: *Text*.
 Поддерживаемые платформы: Linux.

Параметры:

- **regexp** - *all* (по умолчанию) или регулярное выражение;
- **manager** - *all* (по умолчанию) или менеджер пакетов;
- **format** - *full* (по умолчанию) или *short*.

Комментарии:

- Выводит (в алфавитном порядке) установленные пакеты, имя которых соответствует заданному регулярному выражению (*all* выводит все).
- Поддерживаемые менеджеры пакетов (выполняемая команда):
  - pkg (dpkg --get-selections);
  - pkgtool (ls /var/log/packages);
  - rpm (rpm -qa);
  - pacman (pacman -Q);
  - portage.
- Если format задан как *full*, пакеты группируются по менеджерам пакетов (каждый менеджер на отдельной строке, начинающейся с его имени в квадратных скобках).
- Если format задан как *short*, пакеты не группируются и выводятся в одной строке.

Пример:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Подробный список установленных пакетов.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux.

Параметры:

- **regexp** - *all* (по умолчанию) или регулярное выражение;
- **manager** - *all* (по умолчанию) или менеджер пакетов (возможные значения: *rpm*, *dpkg*, *pkgtool*, *pacman* или *portage*).

Комментарии:

- Возвращает неформатированный JSON с установленными пакетами, имя которых соответствует заданному регулярному выражению.
- Вывод представляет собой массив объектов, каждый из которых содержит следующие ключи: name, manager, version, size, architecture, buildtime и installtime (см. подробности).

system.swap.in[<device>,<type>]

 Статистика swap-in (из устройства в память).
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, OpenBSD.

Параметры:

- **device** - укажите устройство, используемое для подкачки (только Linux) или *all* (по умолчанию);
- **type** - возможные значения:
  - *count* (количество операций swap-in, по умолчанию на платформах, отличных от Linux);
  - *sectors* (секторы, считанные в swap);
  - *pages* (страницы, считанные в swap, по умолчанию на Linux).

Комментарии:

- Источником этой информации являются:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Обратите внимание, что *pages* будет работать только если device не указан.
- Параметр типа *sectors* поддерживается только на Linux.

Пример:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Статистика swap-out (из памяти на устройство).
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, OpenBSD.

Параметры:

- **device** - укажите устройство, используемое для swap, (только Linux) или *all* (по умолчанию);
- **type** - возможные значения:
  - *count* (количество операций swapout, по умолчанию на платформах, отличных от Linux);
  - *sectors* (секторы, выгруженные в swap);
  - *pages* (страницы, выгруженные в swap, по умолчанию на Linux).

Комментарии:

- Источником этой информации являются:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Обратите внимание, что *pages* будет работать только если устройство не указано.
- Параметр типа *sectors* поддерживается только на Linux.

Пример:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Размер пространства подкачки в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* - для байтов; *Float* - для процентов.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Параметры:

- **device** - укажите устройство, используемое для подкачки (только FreeBSD) или *all* (по умолчанию);
- **type** - возможные значения:
  - *free* (свободное пространство подкачки, по умолчанию);
  - *pfree* (свободное пространство подкачки, в процентах);
  - *pused* (используемое пространство подкачки, в процентах);
  - *total* (общее пространство подкачки);

- *used* (используемое пространство подкачки).

Комментарии:

- Обратите внимание, что *pfree*, *pused* не поддерживаются в Windows, если размер swap равен 0.
- Если device не указан, агент Zabbix будет учитывать только устройства подкачки (файлы), физическая память будет проигнорирована. Например, в системах Solaris команда `swap -s` включает часть физической памяти и устройства подкачки (в отличие от `swap -l`).

Пример:

```
system.swap.size[,pfree] →
```

system.uname

 Идентификация системы.
 Возвращаемое значение: *Строка*.
 Смотрите Поддерживаемые платформы.

Комментарии:

- В Unix значение этого элемента данных извлекается при помощи системного вызова uname();
- Этот элемент данных в Windows возвращает архитектуру ОС, тогда как в Unix он возвращает архитектуру CPU.

Примеры:

```
system.uname → FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
system.uname → Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
```

system.uptime

 Время работы в секундах.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Для получения читабельных значений используйте при настройке элемента данных единицы измерения **s** или **uptime**.

system.users.num

 Количество пользователей, находящихся в системе.
 Возвращаемое значение: *Целое число*.
 Смотрите Поддерживаемые платформы.

Для получения значений используется команда **who** на стороне агента.

vfs.dev.discovery

 Список блочных устройств и их тип. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux.

vfs.dev.get[<mode>,<regexp>]

 Список блочных устройств с подробной информацией. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, AIX.

Параметры:

- **mode** - возможные значения:
  - *disks* (по умолчанию) - возвращает список дисков;
  - *disk_stats* - возвращает статистику для каждого диска;
  - *devices* (только Linux) - возвращает список блочных устройств (диски и разделы);
  - *device_stats* (только Linux) - возвращает статистику для каждого устройства (диски и разделы).
- **regexp** - регулярное выражение, используемое для фильтрации блочных устройств по их имени.

vfs.dev.read[<device>,<type>,<mode>]

 Статистика чтения диска.
 Возвращаемое значение: - *Integer* - при `type` в значениях *sectors*, *operations*, *bytes*; - *Float* - при `type` в значениях *sps*, *ops*, *bps*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Параметры:

- **device** - дисковое устройство (по умолчанию *all* [3]);
- **type** - возможные значения: *sectors*, *operations*, *bytes*, *sps*, *ops* или *bps* (*sps*, *ops*, *bps* означают соответственно: sectors, operations, bytes per second);
- **mode** - возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*. Этот параметр поддерживается только при `type` в значениях: sps, ops, bps.

Комментарии:

- Если используется интервал обновления три часа или более[2], этот элемент данных всегда будет возвращать 0.

- Параметры типа *sectors* и *sps* поддерживаются только в Linux.
- Параметр типа *ops* поддерживается только в Linux и FreeBSD.
- Параметр типа *bps* поддерживается только в FreeBSD.
- Параметр типа *bytes* поддерживается только в FreeBSD, Solaris, AIX, OpenBSD.
- Параметр `mode` поддерживается только в Linux, FreeBSD.
- Можно использовать относительные имена устройств (например, `sda`), а также необязательный префикс `/dev/` (например, `/dev/sda`).
- Поддерживаются логические тома LVM.
- Значения по умолчанию параметра 'type' для разных ОС:
  - *AIX* - operations;
  - *FreeBSD* - bps;
  - *Linux* - sps;
  - *OpenBSD* - operations;
  - *Solaris* - bytes.
- *sps*, *ops* и *bps* на поддерживаемых платформах ограничены 1024 устройствами (1023 отдельными и одним для *all*).

Пример:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Статистика записи на диск.
 Возвращаемое значение: *Integer* - при `type` в значениях *sectors*, *operations*, *bytes*; *Float* - при `type` в значениях *sps*, *ops*, *bps*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Параметры:

- **device** - устройство диска (по умолчанию *all* [3]);
- **type** - возможные значения: *sectors*, *operations*, *bytes*, *sps*, *ops* или *bps* (*sps*, *ops*, *bps* означают соответственно: sectors, operations, bytes per second);
- **mode** - возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*. Этот параметр поддерживается только при `type` в значениях: sps, ops, bps.

Комментарии:

- Если используется интервал обновления три часа или более[2], этот элемент данных всегда будет возвращать 0.
- Параметры типа *sectors* и *sps* поддерживаются только в Linux.
- Параметр типа *ops* поддерживается только в Linux и FreeBSD.
- Параметр типа *bps* поддерживается только в FreeBSD.
- Параметр типа *bytes* поддерживается только в FreeBSD, Solaris, AIX, OpenBSD.
- Параметр `mode` поддерживается только в Linux, FreeBSD.
- Можно использовать относительные имена устройств (например, `sda`), а также необязательный префикс `/dev/` (например, `/dev/sda`).
- Поддерживаются логические тома LVM.
- Значения по умолчанию параметра `type` для разных ОС:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* и *bps* на поддерживаемых платформах ограничены 1024 устройствами (1023 отдельными и одним для *all*).

Пример:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 Количество записей в каталоге.
 Возвращаемое значение: *Integer*.
 См. поддерживаемые платформы.

Параметры:

- **dir** - абсолютный путь к каталогу;
- **regex incl** - регулярное выражение, описывающее шаблон имени сущности (файла, каталога, символической ссылки), которую необходимо включить; если параметр пуст, включаются все сущности (значение по умолчанию);
- **regex excl** - регулярное выражение, описывающее шаблон имени сущности (файла, каталога, символической ссылки), которую необходимо исключить; если параметр пуст, сущности не исключаются (значение по умолчанию);
- **types incl** - типы записей каталога, которые необходимо подсчитывать. Возможные значения:

  - *file* - обычный файл;
  - *dir* - подкаталог;
  - *sym* - символическая ссылка;
  - *sock* - сокет;
  - *bdev* - блочное устройство;
  - *cdev* - символьное устройство;
  - *fifo* - FIFO;
  - *dev* - синоним "bdev,cdev";
  - *all* - все типы (по умолчанию), то есть "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов необходимо разделять запятыми и заключать в кавычки.
- **types excl** - типы записей каталога (см. types incl), которые НЕ нужно подсчитывать. Если тип записи указан одновременно в types incl и types excl, записи каталога этого типа НЕ подсчитываются.
- **max depth** - максимальная глубина вложенности подкаталогов для обхода:
  - **-1** (по умолчанию) - без ограничений;
  - **0** - не переходить в подкаталоги.
- **min size** - минимальный размер (в байтах) файла, который необходимо подсчитывать. Файлы меньшего размера не подсчитываются. Можно использовать суффиксы размера памяти.
- **max size** - максимальный размер (в байтах) файла, который необходимо подсчитывать. Файлы большего размера не подсчитываются. Можно использовать суффиксы размера памяти.
- **min age** - минимальный возраст (в секундах) записи каталога, которую необходимо подсчитывать. Более новые записи не подсчитываются. Можно использовать суффиксы времени.
- **max age** - максимальный возраст (в секундах) записи каталога, которую необходимо подсчитывать. Записи такого возраста и старше не подсчитываются (по времени изменения). Можно использовать суффиксы времени.
- **regex excl dir** - регулярное выражение, описывающее шаблон имени каталога, который необходимо исключить. Все содержимое каталога будет исключено (в отличие от regex_excl).

Комментарии:

- В Windows этот элемент данных поддерживает пути UNC, которые могут представлять собой угрозу безопасности.
- Переменные окружения, например %APP_HOME%, $HOME и %TEMP%, не поддерживаются.
- Псевдокаталоги ”.” и ”..” никогда не подсчитываются.
- Символические ссылки никогда не используются для обхода каталогов.
- При подсчете количества записей к файлам и каталогам применяются оба параметра regex incl и regex excl, но они игнорируются при выборе подкаталогов для обхода (если regex incl имеет значение (?i)^.+\.zip$, а max depth не задан, будут обойдены все подкаталоги, но подсчитаны только файлы типа zip).
- Время выполнения ограничено значением тайм-аута в конфигурации агента (3 секунды). Поскольку обход большого каталога может занять больше времени, данные возвращены не будут, а элемент данных станет неподдерживаемым. Частичный результат подсчета возвращен не будет.
- При фильтрации по размеру значимый размер имеют только обычные файлы. В Linux и BSD каталоги также имеют ненулевой размер (обычно несколько КБ). Размеры устройств равны нулю, например размер /dev/sda1 не отражает размер соответствующего раздела. Поэтому при использовании <min_size> и <max_size> рекомендуется указывать <types_incl> как *”file”*, чтобы избежать неожиданных результатов.

Примеры:

```
vfs.dir.count[/dev] #                       /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #
```

vfs.dir.get[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 Список записей каталога.
 Возвращаемое значение: *строка JSON*.
 См. поддерживаемые платформы.

Параметры:

- **dir** - абсолютный путь к каталогу;
- **regex incl** - регулярное выражение, описывающее шаблон имени сущности (файла, каталога, символической ссылки), которую необходимо включить; если параметр пуст, включаются все сущности (значение по умолчанию);
- **regex excl** - регулярное выражение, описывающее шаблон имени сущности (файла, каталога, символической ссылки), которую необходимо исключить; если параметр пуст, сущности не исключаются (значение по умолчанию);
- **types incl** - типы записей каталога, которые необходимо включить в список. Возможные значения:
  - *file* - обычный файл;
  - *dir* - подкаталог;
  - *sym* - символическая ссылка;
  - *sock* - сокет;
  - *bdev* - блочное устройство;
  - *cdev* - символьное устройство;

    - *fifo* - FIFO;
    - *dev* - синоним "bdev,cdev";
    - *all* - все типы (по умолчанию), то есть "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов необходимо разделять запятыми и заключать в кавычки.
- **types excl** - типы записей каталога (см. types incl), которые НЕ нужно включать в список. Если тип записи указан одновременно в types incl и types excl, записи каталога этого типа НЕ включаются в список.
- **max depth** - максимальная глубина обхода подкаталогов:
    - **-1** (по умолчанию) - без ограничений;
    - **0** - не переходить в подкаталоги.
- **min size** - минимальный размер (в байтах) файла, который необходимо включить в список. Файлы меньшего размера не включаются в список. Можно использовать суффиксы размера памяти.
- **max size** - максимальный размер (в байтах) файла, который необходимо включить в список. Файлы большего размера не включаются в список. Можно использовать суффиксы размера памяти.
- **min age** - минимальный возраст (в секундах) записи каталога, которую необходимо включить в список. Более новые записи не включаются в список. Можно использовать суффиксы времени.
- **max age** - максимальный возраст (в секундах) записи каталога, которую необходимо включить в список. Записи с таким или большим возрастом не включаются в список (по времени изменения). Можно использовать суффиксы времени.
- **regex excl dir** - регулярное выражение, описывающее шаблон имени каталога, который необходимо исключить. Все содержимое каталога будет исключено (в отличие от regex excl).

Комментарии:

- В Windows этот элемент данных поддерживает пути UNC, которые могут представлять угрозу безопасности.
- Переменные окружения, например %APP_HOME%, $HOME и %TEMP%, не поддерживаются.
- Псевдокаталоги "." и ".." никогда не включаются в список.
- Символические ссылки никогда не используются для обхода каталогов.
- regex incl и regex excl применяются к файлам и каталогам при формировании списка записей, но игнорируются при выборе подкаталогов для обхода (если regex incl имеет значение (?i)^.+\.zip$, а max depth не задан, будут обработаны все подкаталоги, но учитываться будут только файлы типа zip).
- Время выполнения ограничено значением параметра timeout в конфигурации агента. Поскольку обход большого каталога может занять больше времени, данные возвращены не будут, а элемент данных станет неподдерживаемым. Частичный список возвращен не будет.
- При фильтрации по размеру значимый размер имеют только обычные файлы. В Linux и BSD каталоги также имеют ненулевой размер (обычно несколько КБ). Размер устройств равен нулю, например размер /dev/sda1 не отражает размер соответствующего раздела. Поэтому при использовании min size и max size рекомендуется указывать types incl как "*file*", чтобы избежать неожиданных результатов.

Примеры:

```
vfs.dir.get[/dev] #                  /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 Размер каталога (в байтах).
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, Windows. Элемент данных может работать и на других UNIX-подобных платформах.

Параметры:

- **dir** - абсолютный путь к каталогу;
- **regex incl** - регулярное выражение, описывающее шаблон имени объекта (файл, каталог, символическая ссылка), который нужно включить; если пусто, включаются все (значение по умолчанию);
- **regex excl** - регулярное выражение, описывающее шаблон имени объекта (файл, каталог, символическая ссылка), который нужно исключить; если пусто, ничего не исключается (значение по умолчанию);
- **mode** - возможные значения:
    - *apparent* (по умолчанию) - получает видимые размеры файлов, а не использование диска (действует как du -sb dir);
    - *disk* - получает использование диска (действует как du -s -B1 dir). В отличие от команды du, элемент данных vfs.dir.size учитывает скрытые файлы при вычислении размера каталога (действует как du -sb .[^.]* * внутри dir).
- **max depth** - максимальная глубина обхода подкаталогов:
    - **-1** (по умолчанию) - без ограничений;
    - **0** - без перехода в подкаталоги.
- **regex excl dir** - регулярное выражение, описывающее шаблон имени каталога, который нужно исключить. Все содержимое каталога будет исключено (в отличие от regex excl)

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может быть риском безопасности.
- Учитываются только каталоги, для которых у пользователя *zabbix* есть как минимум право на чтение. Для каталогов только с правом на чтение вычисляется размер самого каталога. Каталоги с правами на чтение и выполнение вычисляются вместе с содержимым.
- Для больших каталогов или медленных дисков этот элемент данных может завершиться по тайм-ауту из-за настройки Timeout в конфигурационных файлах агента и сервера/прокси. При необходимости увеличьте значения тайм-аута.
- Ограничение размера файла зависит от поддержки больших файлов.

Примеры:

```
vfs.dir.size[/tmp,log] #                    /tmp,       'log'
vfs.dir.size[/tmp,log,^.+\.old$] #                   /tmp,       'log'          ,               ,
```

vfs.file.cksum[file,<mode>]

 Контрольная сумма файла, вычисленная по алгоритму UNIX cksum.
 Возвращаемое значение: *Integer* — при `mode` = *crc32*, *String* — при `mode` = *md5*, *sha256*.
 См. поддерживаемые платформы.

Параметры:

- **file** — полный путь к файлу;
- **mode** — *crc32* (по умолчанию), *md5* или *sha256*.

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может представлять риск безопасности.
- Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.cksum[/etc/passwd]
```

Примеры возвращаемых значений (соответственно crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Получение содержимого файла[7].
 Возвращаемое значение: *Text*.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **encoding** - идентификатор кодовой страницы identifier.

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может представлять риск безопасности.
- Возвращаемое значение ограничено 16 МБ (включая завершающие пробельные символы, которые обрезаются); также действуют ограничения базы данных.
- Пустая строка возвращается, если файл пуст или содержит только символы LF/CR.
- Маркер порядка байтов (BOM) исключается из вывода.

Пример:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Проверяет, существует ли файл.
 Возвращаемое значение: 0 - не найден; 1 - файл указанного типа существует.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **types incl** - список типов файлов для включения, возможные значения:
  - *file* (обычный файл, по умолчанию (если `types_excl` не задан));
  - *dir* (каталог);
  - *sym* (символическая ссылка);
  - *sock* (сокет);

  - *bdev* (блочное устройство);
  - *cdev* (символьное устройство);
  - *fifo* (FIFO);
  - *dev* (синоним для "bdev,cdev");
  - *all* (все перечисленные типы, по умолчанию, если задан `types_excl`).
- **types excl** - список типов файлов для исключения, см. `types_incl` для возможных значений (по умолчанию типы не исключаются)

Комментарии:

- В Windows этот элемент следует по UNC-путям, что может представлять риск безопасности.
- Несколько типов должны быть разделены запятыми, а весь набор заключен в кавычки "".
- Если один и тот же тип указан и в <types_incl>, и в <types_excl>, файлы этого типа исключаются.
- Ограничение размера файла зависит от поддержки больших файлов.

Примеры:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 Возвращает информацию о файле.
 Возвращаемое значение: *JSON string*.
 См. поддерживаемые платформы.

Параметр:

- **file** - полный путь к файлу

Комментарии:

- В Windows этот элемент данных следует по путям UNC, что может представлять риск безопасности.
- Поддерживаемые типы файлов в UNIX-подобных системах: обычный файл, каталог, символическая ссылка, сокет, блочное устройство, символьное устройство, FIFO.
- Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 MD5-сумма файла.
 Возвращаемое значение: символьная строка (MD5-хэш файла).
 См. поддерживаемые платформы.

Параметр:

- **file** - полный путь к файлу

Комментарии:

- В Windows этот элемент данных следует по путям UNC, что может представлять риск безопасности.
- Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Пример возвращаемого значения:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Получает владельца файла.
 Возвращаемое значение: *String*.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **ownertype** - *user* (по умолчанию) или *group* (только Unix);
- **resulttype** - *name* (по умолчанию) или *id*; для id - возвращает uid/gid в Unix, SID в Windows.

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может быть риском безопасности.
- Ограничение на размер файла зависит от поддержки больших файлов.

Примеры:

```
vfs.file.owner[/tmp/zabbix_server.log] #                /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #       ID           /tmp/zabbix_server.log
```

vfs.file.permissions[файл]

 Получение строки из 4 цифр, содержащей восьмеричное число с правами доступа Unix.
 Возвращаемое значение: *Строка*.
 Поддерживаемые платформы: Linux. Элемент данных может работать на других UNIX-подобных платформах.

Параметры:

- **файл** — абсолютный путь к файлу

Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.permissions[/etc/passwd] #                /etc/passwd,      , «0644»
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Получить строку в файле[7].
 Возвращаемое значение: строка, содержащая совпавшую строку, или значение, указанное с помощью необязательного параметра `output`.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **regexp** - регулярное выражение, описывающее требуемый шаблон;
- **encoding** - идентификатор кодовой страницы;
- **start line** - номер первой строки для поиска (по умолчанию первая строка файла);
- **end line** - номер последней строки для поиска (по умолчанию последняя строка файла);
- **output** - необязательный шаблон форматирования вывода. Последовательность экранирования **\0** заменяется совпавшей частью текста (от первого символа, где начинается совпадение, до символа, где оно заканчивается), а последовательность экранирования **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает число захваченных групп).

Комментарии:

- В Windows этот элемент данных следует UNC-путям, что может представлять риск безопасности.
- Ограничение размера файла зависит от поддержки больших файлов.
- Возвращается только первая совпавшая строка.
- Если ни одна строка не соответствует выражению, возвращается пустая строка.
- Метка порядка байтов (BOM) исключается из вывода.
- Извлечение содержимого с использованием параметра `output` выполняется на агенте.

Примеры:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] →       ID         *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Найти строку в файле[7].
 Возвращаемые значения: 0 - совпадение не найдено; 1 - найдено.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **regexp** - регулярное выражение, описывающее требуемый шаблон;
- **encoding** - идентификатор кодовой страницы кодировки;
- **start line** - номер первой строки для поиска (по умолчанию первая строка файла);
- **end line** - номер последней строки для поиска (по умолчанию последняя строка файла).

Комментарии:

- В Windows этот элемент данных следует UNC-путям, что может представлять риск безопасности.
- Ограничение размера файла зависит от поддержки больших файлов.
- Метка порядка байтов (BOM) игнорируется.

Пример:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 Размер файла (в байтах).
 Возвращаемое значение: *Integer*.
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **mode** - возможные значения:
  - *bytes* (по умолчанию);
  - *lines* (пустые строки тоже учитываются).

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может быть риском безопасности.
- Файл должен иметь права на чтение для пользователя *zabbix*.
- Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Информация о времени файла.
 Возвращаемое значение: *Integer* (Unix timestamp).
 См. поддерживаемые платформы.

Параметры:

- **file** - полный путь к файлу;
- **mode** - возможные значения:
  - *modify* (по умолчанию) - время последнего изменения содержимого файла (по умолчанию);
  - *access* - время последнего чтения файла;
  - *change* - время последнего изменения свойств файла.

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может быть риском безопасности.
- Ограничение размера файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 Список смонтированных файловых систем с их типом и параметрами монтирования. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get[<mode>,<mountpoint>]

 Список смонтированных файловых систем с указанием их типа и параметров монтирования, а в режиме *full* — доступного дискового пространства и статистики inode. Можно использовать для обнаружения низкого уровня.
 Возвращаемое значение: *строка JSON*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Параметры:

- **mode** — возможные значения:
  - *full* (по умолчанию) — свойства файловой системы, а также статистика дискового пространства и inode;
  - *short* — только свойства файловой системы, без статистики дискового пространства и inode;
- **mountpoint** — точка монтирования, для которой необходимо вернуть данные (например, `/var` или `C:`). Если параметр не указан, возвращаются все смонтированные файловые системы.

Комментарии:

- В режиме *short* возвращаются только имя файловой системы, тип и параметры монтирования (в Windows: имя, тип, метка и тип диска). Статистика файловых систем не запрашивается; в Unix-подобных системах считывается только список смонтированных файловых систем (например, `/proc/mounts` в Linux), поэтому проверка не вызывает автомонтирование и не зависит от не отвечающих сетевых файловых систем. Это делает режим *short* подходящим для обнаружения низкого уровня.

- Значение **mountpoint** должно в точности соответствовать свойству `fsname`. В Windows сравнение выполняется без учета регистра; указывайте точку монтирования без завершающей обратной косой черты (например, `C:`).
- Если параметр **mountpoint** указан, данные возвращаются только для этой точки монтирования. Если смонтированная файловая система не найдена, возвращается пустой массив (`[]`).
- Выходные данные могут содержать несколько записей с одинаковым значением `fsname`, если в одном каталоге смонтировано несколько файловых систем.
- В режиме *full* доступное дисковое пространство и статистика inode возвращаются только в том случае, если пользователь агента Zabbix имеет разрешение на обход каждого каталога в пути к точке монтирования; в противном случае файловая система исключается из выходных данных. Это не относится к режиму *short*, поэтому в нем могут отображаться файловые системы, отсутствующие в выходных данных режима *full*.
- Свойства `fslabel` и `fsdrivetype` возвращаются только в Windows; свойства `inodes` и `options` в Windows не возвращаются.
- Также отображаются файловые системы с количеством inode, равным нулю, что может быть характерно для файловых систем с динамическими inode (например, btrfs).
- См. также: Обнаружение смонтированных файловых систем.

Примеры:

```
vfs.fs.get #                              ;     ,      vfs.fs.get[full]
vfs.fs.get[short] #
vfs.fs.get[full,/var] #                          /var
vfs.fs.get[short,C:] #             C: (Windows)
```

vfs.fs.inode[fs,<режим>]

 Количество или процент inodes.
 Возвращаемое значение: *Целое число* — для количества; *Число с плавающей точной* — для процентов.
 Смотрите Поддерживаемые платформы.

Параметры:

- **fs** — файловая система;
- **режим** — возможные значения: *total* (по умолчанию), *free*, *used*, *pfree* (свободно, в процентах) или *pused* (использовано, в процентах).

Если количество inode'ов станет нулевым, что может случаться на файловых системах с динамическим количеством файловых дескрипторов (например, btrfs), значения pfree/pused будут сообщаться как «100» и «0» соответственно.

Пример:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<режим>]

 Дисковое пространство в байтах или процентах от общего размера.
 Возвращаемое значение: *Целое число* — для байт; *Число с плавающей точкой* — для процентов.
 Смотрите Поддерживаемые платформы.

Параметры:

- **fs** — файловая система;
- **режим** — возможные значения: *total* (по умолчанию), *free*, *used*, *pfree* (доступно, в процентах) или *pused* (использовано, в процентах).

Комментарии:

- Если файловая система не примонтирована, будет возвращён размер локальной файловой системы, где расположена точка монтирования;
- Зарезервированное место на файловой системе принимается во внимание и не включено при использовании режима *free*.

Пример:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 Размер памяти в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* - для байтов; *Float* - для процентов.
 См. поддерживаемые платформы.

Параметр:

- **mode** - возможные значения: *total* (по умолчанию), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (использовано, в процентах), *available* или *pavailable* (доступно, в процентах).

Комментарии:

- Этот элемент данных принимает три категории параметров:
  - *total* - общий объем памяти;
  - типы памяти, зависящие от платформы: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - оценки на уровне пользователя о том, сколько памяти используется и доступно: *used*, *pused*, *available*, *pavailable*.
- Параметр режима *active* поддерживается только в FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Параметры режима *anon*, *exec*, *file* поддерживаются только в NetBSD.
- Параметр режима *buffers* поддерживается только в Linux, FreeBSD, OpenBSD, NetBSD.
- Параметр режима *cached* поддерживается только в Linux, FreeBSD, AIX, OpenBSD, NetBSD.
- Параметры режима *inactive*, *wired* поддерживаются только в FreeBSD, MacOS X, OpenBSD, NetBSD.
- Параметр режима *pinned* поддерживается только в AIX.
- Параметр режима *shared* поддерживается только в Linux 2.4, FreeBSD, OpenBSD, NetBSD.
- См. также дополнительные сведения для этого элемента данных.

Пример:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>,<redirect_limit>]

 Получение содержимого веб-страницы.
 Возвращаемое значение: исходный текст веб-страницы (включая заголовки).
 См. поддерживаемые платформы.

Параметры:

- **host** - имя узла сети или URL (в формате `scheme://host:port/path`, где обязательным является только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, используется *http*. Если указан URL, параметры `path` и `port` должны быть пустыми. Указание имени пользователя и пароля при подключении к серверам, требующим аутентификации, например `http://user:password@www.example.com`, возможно только при поддержке cURL [4]. В именах узлов сети поддерживается Punycode.
- **path** - путь к HTML-документу (по умолчанию /);
- **port** - номер порта (по умолчанию 80 для HTTP);
- **redirect_limit** - максимальное количество перенаправлений, которым разрешено следовать [4], от 0 до 1000 (по умолчанию 0 - перенаправления не обрабатываются).

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен либо если `redirect_limit` больше 0 и количество перенаправлений превышает это значение.
- `host` может быть именем узла сети, доменным именем, адресом IPv4 или IPv6. Однако для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.
- Если перенаправления обрабатываются, возвращаемое значение содержит заголовки каждого ответа на перенаправление, за которыми следуют заголовки и тело окончательного ответа.

Примеры:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
web.page.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 Время загрузки полной веб-страницы (в секундах).
 Возвращаемое значение: *Float*.
 См. поддерживаемые платформы.

Параметры:

- **host** - имя узла сети или URL (в формате `scheme://host:port/path`, где обязательным является только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, используется *http*. Если указан URL, параметры `path` и `port` должны быть пустыми. Указание имени пользователя и пароля при подключении к серверам, требующим аутентификации, например `http://user:password@www.example.com`, возможно только при поддержке cURL [4]. В именах узлов сети поддерживается Punycode.
- **path** - путь к HTML-документу (по умолчанию /);
- **port** - номер порта (по умолчанию 80 для HTTP);

- **redirect_limit** - максимальное количество перенаправлений, за которыми следует перейти [4], от 0 до 1000 (по умолчанию 0 - перенаправления не обрабатываются).

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен, либо если `redirect_limit` больше 0 и количество перенаправлений превышает это значение.
- `host` может быть именем узла сети, доменным именем, адресом IPv4 или IPv6. Однако для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.
- Если перенаправления обрабатываются, время загрузки включает время, затраченное на все перенаправления.

Примеры:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Поиск строки на веб-странице.
 Возвращаемое значение: найденная строка или значение, указанное необязательным параметром `output`.
 См. поддерживаемые платформы.

Параметры:

- **host** — имя узла сети или URL (в формате `scheme://host:port/path`, где обязательным является только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, используется *http*. Если указан URL, параметры `path` и `port` должны быть пустыми. Указание имени пользователя и пароля при подключении к серверам, требующим аутентификации, например `http://user:password@www.example.com`, возможно только при поддержке cURL [4]. В именах узлов сети поддерживается Punycode.
- **path** — путь к HTML-документу (по умолчанию /).
- **port** — номер порта (по умолчанию 80 для HTTP).
- **regexp** — регулярное выражение, описывающее требуемый шаблон.
- **length** — максимальное количество возвращаемых символов;
- **output** — необязательный шаблон форматирования результата. Экранирующая последовательность **\0** заменяется найденной частью текста (от первого символа, с которого начинается совпадение, до символа, на котором оно заканчивается), а экранирующая последовательность **\N** (где N=1...9) заменяется N-й найденной группой (или пустой строкой, если N превышает количество захваченных групп).
- **redirect_limit** — максимальное количество перенаправлений, за которыми необходимо следовать [4], от 0 до 1000 (по умолчанию 0 — перенаправления не обрабатываются).

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен, либо если `redirect_limit` больше 0 и количество перенаправлений превышает это значение.
- `host` может быть именем узла сети, доменным именем, адресом IPv4 или IPv6. Однако для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.
- Извлечение содержимого с помощью параметра `output` выполняется на агенте.
- Если выполняется переход по перенаправлениям, заголовки каждого ответа на перенаправление также проверяются.

Примеры:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 Метаданные узла сети агента.
 Возвращаемое значение: *String*.
 См. поддерживаемые платформы.

Возвращает значение параметров `HostMetadata` или `HostMetadataItem`, либо пустую строку, если они не заданы.

agent.hostname

 Имя узла сети агента.
 Возвращаемое значение: *String*.
 См. поддерживаемые платформы.

Возвращает:

- При пассивной проверке — имя первого узла сети, указанного в параметре Hostname файла конфигурации агента;
- При активной проверке — имя текущего узла сети.

agent.ping

 Проверка доступности агента.
 Возвращаемое значение: ничего - недоступен; 1 - доступен.
 См. поддерживаемые платформы.

Используйте функцию триггера `nodata()` для проверки недоступности узла сети.

agent.variant

 Проверка разновидности агента (Zabbix агент или Zabbix агент 2).
 Возвращаемое значение: 1 — Zabbix агент; 2 — Zabbix агент 2.
 Смотрите Поддерживаемые платформы.

agent.version

 Версия Zabbix агента.
 Возвращаемое значение: *Строка*.
 Смотрите Поддерживаемые платформы.

Пример возвращаемого значения:

```
6.0.3
```

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

 Возвращает набор внутренних метрик сервера Zabbix или прокси. Если указаны `ip` и `port`, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.
 Возвращаемое значение: *JSON string*.
 См. поддерживаемые платформы.

Параметры:

- **ip** - список IP/DNS/сетевых масок серверов/прокси, к которым будет выполнен удаленный запрос (по умолчанию `127.0.0.1`);
- **port** - порт сервера/прокси, к которому будет выполнен удаленный запрос (по умолчанию 10051)

Комментарии:

- Элемент данных возвращает выбранный набор внутренних метрик. Подробности см. в разделе Удаленный мониторинг статистики Zabbix.
- Обратите внимание, что запрос статистики будет принят только с адресов, перечисленных в параметре `StatsAllowedIP` сервер/прокси на целевом экземпляре.

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

 Возвращает количество контролируемых элементов данных в очереди, задержанных на сервере Zabbix или прокси. Если указаны `ip` и `port`, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.
 Возвращаемое значение: *JSON string*.
 См. поддерживаемые платформы.

Параметры:

- **ip** - список IP/DNS/сетевых масок серверов/прокси, к которым выполняется удаленный запрос (по умолчанию `127.0.0.1`);
- **port** - порт сервера/прокси, к которому выполняется удаленный запрос (по умолчанию 10051)
- **queue** - константа (используется как есть)
- **from** - задержка не менее чем на (по умолчанию 6 секунд)
- **to** - задержка не более чем на (по умолчанию бесконечность)

Обратите внимание, что запрос stats будет принят только с адресов, перечисленных в параметре 'StatsAllowedIP' server/proxy на целевом экземпляре.

Сноски

[1] Примечание, относящееся к Linux. Агент Zabbix должен иметь доступ только для чтения к файловой системе `/proc`. Патчи ядра с сайта www.grsecurity.org ограничивают права доступа непривилегированных пользователей.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: агент Zabbix завершит подключения к «устаревшим» устройствам, если значения элементов данных не запрашивались более 3 часов. Это может произойти, если в системе есть устройства с динамически изменяющимися путями или если устройство было удалено вручную. Также обратите внимание, что эти элементы данных при использовании интервала обновления 3 часа или более всегда будут возвращать значение «0».

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: если для первого параметра используется значение `all` по умолчанию, ключ вернет сводную статистику, включая все блочные устройства, такие как sda и sdb, их разделы (sda1, sda2, sdb3...), составные устройства (RAID MD), созданные на основе этих блочных устройств и разделов, а также логические тома (LVM), созданные на основе этих блочных устройств и разделов. В таких случаях возвращаемые значения следует рассматривать только как относительные (динамически изменяющиеся во времени), но не как абсолютные.

[4] HTTPS и переходы по перенаправлениям (`redirect_limit` больше 0) поддерживаются только в том случае, если агент Zabbix скомпилирован с поддержкой cURL; в противном случае элемент данных становится неподдерживаемым. Агент Zabbix 2 не требует cURL ни для одной из этих функций.

[5] Значения bytes и errors не поддерживаются для интерфейсов loopback в системах Solaris вплоть до Solaris 10 6/06 включительно, поскольку статистика байтов, ошибок и использования не сохраняется и/или не предоставляется ядром. Однако при мониторинге системы Solaris через net-snmp значения могут возвращаться, поскольку net-snmp содержит устаревший код из cmu-snmp, датируемый 1997 годом. Если этому коду не удается прочитать значения байтов из статистики интерфейса, он возвращает счетчик пакетов (который существует для интерфейсов loopback), умноженный на произвольное значение 308. Это предполагает, что средняя длина пакета составляет 308 октетов, что является очень приблизительной оценкой, поскольку ограничение MTU для интерфейсов loopback в системах Solaris составляет 8892 байта. Не следует считать эти значения корректными или даже близкими к точным. Это приблизительные оценки. Агент Zabbix не выполняет подобных предположений, однако net-snmp будет возвращать значение для этих полей.

[6] Командная строка в Solaris, получаемая из /proc/pid/psinfo, ограничена 80 байтами и содержит командную строку в том виде, в котором она была при запуске процесса.

[7] Элементы данных vfs.file.contents[], vfs.file.regexp[], vfs.file.regmatch[] можно использовать для получения содержимого файлов. Если необходимо ограничить доступ к определенным файлам с конфиденциальной информацией, запустите агент Zabbix от имени пользователя, не имеющего разрешений на просмотр этих файлов.

Использование с утилитами командной строки

Обратите внимание, при тестировании или использовании ключей элементов данных с утилитами zabbix_agentd или zabbix_get из командной строки вы также должны учитывать синтаксис командной оболочки.

Например, если определённый параметр ключа должен быть заключён в двойные кавычки, вам придется явно экранировать эти двойные кавычки, в противном случае они будут обрезаны командной оболочкой как специальные символы и не будут переданы утилите Zabbix.

Примеры:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Настройки кодировки

Чтобы убедиться, что полученные данные не повреждены, можно указать правильную кодировку для обработки проверки (например, vfs.file.contents) в параметре encoding. Список поддерживаемых кодировок (идентификаторов кодовых страниц) можно найти в документации libiconv (проект GNU) или в документации Microsoft Windows SDK для раздела "Code Page Identifiers". Обратите внимание, что Microsoft иногда помечает некоторые кодовые страницы как «доступные только управляемым приложениям», то есть они доступны только через среду выполнения .NET и недоступны через собственные API Win32. Агент Zabbix реализует собственную логику преобразования кодировок, поэтому эти кодовые страницы поддерживаются агентом Zabbix даже в тех случаях, когда собственные функции Windows их не предоставляют.

Если в параметре encoding кодировка не указана, применяются следующие стратегии определения:

- Если кодировка не указана (или указана пустая строка), предполагается, что используется UTF-8, а данные обрабатываются «как есть»;
- Анализ BOM - применяется к элементам данных vfs.file.contents, vfs.file.regexp, vfs.file.regmatch. Предпринимается попытка определить правильную кодировку с помощью метки порядка байтов (BOM) в начале файла. Если BOM отсутствует, вместо этого применяется стандартная стратегия определения (см. выше).

Устранение неполадок с элементами данных агента

В случае пассивных проверок, чтобы элемент данных не остался без значения из-за того, что время ожидания запроса сервера к агенту истекает раньше, следует учитывать следующее:

- Если версия агента старше версии сервера, значение Timeout в настройках элемента данных (или глобальное время ожидания) может потребоваться установить выше, чем значение Timeout в файле конфигурации агента.
- Если версия агента новее версии сервера, значение Timeout в файле конфигурации сервера может потребоваться установить выше, чем значение Timeout в файле конфигурации агента.

Zabbix агент 2

Zabbix agent 2 поддерживает все ключи элементов данных, поддерживаемые для Zabbix agent на Unix и Windows. На этой странице приведены сведения о дополнительных ключах элементов данных, которые можно использовать только с Zabbix agent 2, сгруппированных по плагинам, к которым они относятся.

Ключи элементов данных перечислены без параметров и дополнительной информации. Нажмите на ключ элемента данных, чтобы увидеть полные сведения.

Ключ элемента данных	Описание	Плагин
ceph.df.details	Использование данных кластера и их распределение между пулами.	Ceph (загружаемый)
ceph.osd.stats	Сводная статистика и статистика по каждому OSD.	
ceph.osd.discovery	Список обнаруженных OSD.	
ceph.osd.dump	Пороги использования и состояния OSD.	
ceph.ping	Проверяет, можно ли установить соединение с Ceph.	
ceph.pool.discovery	Список обнаруженных пулов.	
ceph.status	Общее состояние кластера.	
docker.container_info	Низкоуровневая информация о контейнере.	Docker
docker.container_stats	Статистика использования ресурсов контейнера.	
docker.containers	Возвращает список контейнеров.	
docker.containers.discovery	Возвращает список контейнеров. Используется для низкоуровневого обнаружения.	
docker.data.usage	Информация о текущем использовании данных.	
docker.images	Возвращает список образов.	
docker.images.discovery	Возвращает список образов. Используется для низкоуровневого обнаружения.	
docker.info	Системная информация.	
docker.ping	Проверяет, работает ли демон Docker.	
ember.get	Возвращает результат для требуемого устройства.	Ember+ (загружаемый)
memcached.ping	Проверяет, активно ли соединение.	Memcached
memcached.stats	Получает вывод команды STATS.	
mongodb.cfg.discovery	Возвращает список обнаруженных серверов конфигурации.	MongoDB (загружаемый)
mongodb.collection.stats	Возвращает различные статистические данные хранилища для указанной коллекции.	
mongodb.collections.discovery	Возвращает список обнаруженных коллекций.	
mongodb.collections.usage	Возвращает статистику использования коллекций.	
mongodb.connpool.stats	Возвращает информацию об открытых исходящих соединениях от текущего экземпляра базы данных к другим участникам шардированного кластера или набора реплик.	
mongodb.db.stats	Возвращает статистику, отражающую состояние указанной базы данных.	
mongodb.db.discovery	Возвращает список обнаруженных баз данных.	
mongodb.jumbo_chunks.count	Возвращает количество jumbo chunks.	
mongodb.oplog.stats	Возвращает состояние набора реплик на основе данных, опрашиваемых из oplog.	
mongodb.ping	Проверяет, активно ли соединение.	
mongodb.rs.config	Возвращает текущую конфигурацию набора реплик.	
mongodb.rs.status	Возвращает состояние набора реплик с точки зрения участника, на котором выполняется метод.	
mongodb.server.status	Возвращает состояние базы данных.	
mongodb.sh.discovery	Возвращает список обнаруженных шардов, присутствующих в кластере.	
mongodb.version	Возвращает версию сервера базы данных.	
mqtt.get	Подписывается на указанный топик или топики (с подстановочными знаками) указанного брокера и ожидает публикаций.	MQTT
mssql.availability.group.get	Возвращает группы доступности.	MSSQL (загружаемый)
mssql.custom.query	Возвращает результат пользовательского запроса.	
mssql.db.get	Возвращает все доступные базы данных MSSQL.	
mssql.job.status.get	Возвращает состояние заданий.	
mssql.last.backup.get	Возвращает время последнего резервного копирования для всех баз данных.	
mssql.local.db.get	Возвращает базы данных, которые участвуют в группе доступности Always On и реплике (основной или вторичной) и расположены на сервере, с которым было установлено соединение.	
mssql.mirroring.get	Возвращает информацию о зеркалировании.	
mssql.nonlocal.db.get	Возвращает базы данных, которые участвуют в группе доступности Always On и реплике (основной или вторичной), расположенные на других серверах (база данных не является локальной для экземпляра SQL Server, с которым было установлено соединение).	
mssql.perfcounter.get	Возвращает счетчики производительности.	

Ключ элемента данных	Описание	Плагин
mssql.ping	Проверяет, активно ли соединение.	
mssql.quorum.get	Возвращает информацию о кворуме.	
mssql.quorum.member.get	Возвращает участников кворума.	
mssql.replica.get	Возвращает реплики.	
mssql.version	Возвращает версию MSSQL.	
mysql.custom.query	Возвращает результат пользовательского запроса.	MySQL
mysql.db.discovery	Возвращает список баз данных MySQL.	
mysql.db.size	Размер базы данных в байтах.	
mysql.get_status_variables	Значения глобальных переменных состояния.	
mysql.ping	Проверяет, активно ли соединение.	
mysql.replication.discovery	Возвращает список репликаций MySQL.	
mysql.replication.get_slave_status	Состояние репликации.	
mysql.version	Версия MySQL.	
net.dns.get	Выполняет DNS-запрос и возвращает подробную информацию о DNS-записи.	Network
nvml.device.count	Количество GPU-устройств.	NVIDIA GPU (загружаемый)
nvml.device.decoder.utilization	Использование декодера GPU-устройства.	
nvml.device.ecc.mode	Режим ECC GPU-устройства.	
nvml.device.encoder.stats.get	Статистика кодировщика GPU-устройства.	
nvml.device.encoder.utilization	Использование кодировщика GPU-устройства.	
nvml.device.energy.consumption	Потребление энергии GPU-устройством.	
nvml.device.errors.memory	Информация об ошибках памяти ECC GPU-устройства.	
nvml.device.errors.register	Информация об ошибках регистров ECC GPU-устройства.	
nvml.device.fan.speed.avg	Средняя скорость вентилятора GPU-устройства.	
nvml.device.get	Обнаруженные GPU с UUID и именем.	
nvml.device.graphics.frequency	Частота графического ядра GPU-устройства.	
nvml.device.memory.bar1.get	Информация о памяти BAR1 GPU-устройства.	
nvml.device.memory.fb.get	Информация о памяти framebuffer GPU-устройства.	
nvml.device.memory.frequency	Частота памяти GPU-устройства.	
nvml.device.pci.utilization	Использование PCI GPU-устройством.	
nvml.device.performance.state	Состояние производительности GPU-устройства.	
nvml.device.power.limit	Ограничение мощности GPU-устройства.	
nvml.device.power.usage	Потребление мощности GPU-устройством.	
nvml.device.serial	Серийный номер GPU-устройства.	
nvml.device.sm.frequency	Частота потокового мультипроцессора GPU-устройства.	
nvml.device.temperature	Температура GPU-устройства.	
nvml.device.utilization	Статистика использования GPU-устройства.	
nvml.device.video.frequency	Частота видеоядра GPU-устройства.	
nvml.system.driver.version	Версия драйвера NVIDIA.	
nvml.version	Версия библиотеки NVML.	
oracle.diskgroups.stats	Возвращает статистику дисковых групп Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Возвращает список дисковых групп ASM.	
oracle.archive.info	Статистика архивных журналов.	
oracle.cdb.info	Информация о контейнерных базах данных (CDB).	
oracle.custom.query	Результат пользовательского запроса.	
oracle.datafiles.stats	Возвращает статистику файлов данных.	
oracle.db.discovery	Возвращает список баз данных.	
oracle.fra.stats	Возвращает статистику Fast Recovery Area (FRA).	
oracle.instance.info	Статистика экземпляра.	
oracle.pdb.info	Информация о подключаемых базах данных (PDB).	
oracle.pdb.discovery	Возвращает список PDB.	
oracle.pga.stats	Возвращает статистику Program Global Area (PGA).	
oracle.ping	Проверяет, можно ли установить соединение с Oracle.	
oracle.proc.stats	Возвращает статистику процессов.	
oracle.redolog.info	Информация о файлах журнала из control file.	
oracle.sga.stats	Возвращает статистику System Global Area (SGA).	
oracle.sessions.stats	Возвращает статистику сеансов.	
oracle.sys.metrics	Возвращает набор значений системных метрик.	
oracle.sys.params	Возвращает набор значений системных параметров.	
oracle.ts.stats	Возвращает статистику табличных пространств.	

Ключ элемента данных	Описание	Плагин
oracle.ts.discovery	Возвращает список табличных пространств.	
oracle.user.info	Возвращает информацию о пользователе Oracle.	
oracle.version	Возвращает версию сервера базы данных.	
pgsql.autovacuum.count	Количество рабочих процессов autovacuum.	PostgreSQL (загружаемый)
pgsql.archive	Информация об архивированных файлах.	
pgsql.bgwriter	Общее количество контрольных точек для кластера баз данных с разбивкой по типу контрольной точки.	
pgsql.cache.hit	Коэффициент попаданий в буферный кэш PostgreSQL.	
pgsql.connections	Возвращает соединения по типам.	
pgsql.custom.query	Возвращает результат пользовательского запроса.	
pgsql.db.age	Возраст самого старого FrozenXID базы данных.	
pgsql.db.bloating_tables	Количество раздувшихся таблиц в каждой базе данных.	
pgsql.db.discovery	Список баз данных PostgreSQL.	
pgsql.db.size	Размер базы данных в байтах.	
pgsql.dbstat	Собирает статистику по каждой базе данных.	
pgsql.dbstat.sum	Сводные данные по всем базам данных в кластере.	
pgsql.locks	Информация о предоставленных блокировках по каждой базе данных.	
pgsql.oldest.xid	Возраст самого старого XID.	
pgsql.ping	Проверяет, активно ли соединение.	
pgsql.queries	Метрики запросов по времени выполнения.	
pgsql.replication.count	Количество резервных серверов.	
pgsql.replication.process	Задержка flush, задержка write и задержка replay для каждого процесса-отправителя.	
pgsql.replication.process.discovery	Обнаружение имен процессов репликации.	
pgsql.replication.recovery_role	Состояние восстановления.	
pgsql.replication.status	Состояние репликации.	
pgsql.replication_lag.b	Задержка репликации в байтах.	
pgsql.replication_lag.sec	Задержка репликации в секундах.	
pgsql.uptime	Время работы PostgreSQL в миллисекундах.	
pgsql.version	Возвращает версию PostgreSQL.	
pgsql.wal.stat	Статистика WAL.	
redis.config	Получает параметры конфигурации экземпляра Redis, соответствующие шаблону.	Redis
redis.info	Получает вывод команды INFO.	
redis.ping	Проверяет, активно ли соединение.	
redis.slowlog.count	Количество записей slow log с момента запуска Redis.	
smart.attribute.discovery	Возвращает список атрибутов устройств S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Возвращает список устройств S.M.A.R.T.	
smart.disk.get	Возвращает все доступные свойства устройств S.M.A.R.T.	
systemd.unit.get	Возвращает все свойства unit systemd.	Systemd
systemd.unit.info	Информация о unit systemd.	
systemd.unit.discovery	Список unit systemd и их сведения.	
web.certificate.get	Проверяет сертификаты и возвращает сведения о сертификате.	Web certificates

Подробности ключей элементов данных

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками **< >**, опциональны.

ceph.df.details[строка_подкл,<пользователь>,<ключ_api>]

 Использование данных кластера и распределение между пулами.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.osd.stats[строка_подкл,<пользователь>,<ключ_api]

 Агрегированная статистика и статистика по каждому OSD.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.osd.discovery[строка_подкл,<пользователь>,<ключ_api]

 Список обнаруженных OSD. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.osd.dump[строка_подкл,<пользователь>,<ключ_api]

 Пороги использования и состояния OSD.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.ping[строка_подкл,<пользователь>,<ключ_api>]

 Проверка, можно ли установить подключение к Ceph.
 Возвращаемое значение: *0* — подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками); *1* — подключение успешно.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.pool.discovery[строка_подкл,<пользователь>,<ключ_api>]

 Список обнаруженных пулов. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

ceph.status[строка_подкл,<пользователь>,<ключ_api>]

 Общее состояние кластера.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, ключ_api** — имя пользователя и ключ API для подключения к Ceph.

docker.container_info[<ID>,<info>]

 Низкоуровневая информация о контейнере.
 Возвращаемое значение: Вывод API вызова ContainerInspect [en], преобразованный в JSON.

Параметры:

- **ID** — ID или имя контейнера;
- **info** — объём возвращаемой информации. Возможные значения: *short* (по умолчанию) или *full*.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.container_stats[<ID>]

 Статистика использования ресурсов контейнером.
 Возвращаемое значение: Вывод API вызова ContainerStats [en] и использование CPU в процентах, преобразованные в JSON.

Параметр:

- **ID** — ID или имя контейнера.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.containers

 Список контейнеров.
 Возвращаемое значение: Вывод API вызова ContainerList [en], преобразованный в JSON.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.containers.discovery[<опции>]

 Получение списка контейнеров. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметр:

- **опции** — задаёт, следует ли обнаруживать все или только запущенные контейнеры. Поддерживаемые значения: *true* — возвращать все контейнеры; *false* — возвращать только запущенные контейнеры (по умолчанию).

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.data.usage

 Информация о текущем использовании данных.
 Возвращаемое значение: Вывод API вызова SystemDataUsage [en], преобразованный в JSON.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.images

 Получение списка образов.
 Возвращаемое значение: Вывод API вызова ImageList [en], преобразованный в JSON.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.images.discovery

 Получение списка образов. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.info

 Информация о системе.
 Возвращаемое значение: Вывод API вызова SystemInfo [en], преобразованный в JSON.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

docker.ping

 Проверка, активен или нет Docker демон.
 Возвращаемое значение: *1* — подключение установлено; *0* — подключение разорвано.

Пользователь Агента 2 («zabbix») должен быть добавлен в группу [en] «docker» для достаточных привилегий. В противном случае проверка завершится с ошибкой.

ember.get[<uri>,<путь>]

 Возвращает результат требуемого устройства.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **uri** — URI устройства Ember+ . По умолчанию: 127.0.0.1:9998;
- **путь** — OID путь к устройству. По умолчанию пусто, возвращает данные корневой коллекции.

memcached.ping[строка_подкл,<пользователь>,<пароль>]

 Проверка, можно ли установить подключение.
 Возвращаемое значение: *1* — подключение активно; *0* — подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками).

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, пароль** — учётные данные для входа в Memchached.

memcached.stats[строка_подкл,<пользователь>,<пароль>,<тип>]

 Получение вывода команды STATS.
 Возвращаемое значение: *JSON* — вывод, преобразованный в JSON.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пользователь, пароль** — учётные данные для входа в Memchached;
- **тип** — тип возвращаемой статистики: *items*, *sizes*, *slabs* или *settings* (по умолчанию пусто, возвращается общая статистика).

mongodb.collection.stats[строка_подкл,<имя_пользователя>,<пароль>,<база_данных>,коллекция]

 Получение различной статистики хранилища по заданной коллекции.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB;
- **база_данных** — имя базы данных (по умолчанию: admin);
- **коллекция** — имя коллекции.

mongodb.cfg.discovery[connString,<user>,<password>]

 Возвращает список обнаруженных серверов конфигурации.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в MongoDB.

mongodb.collections.discovery[строка_подкл,<имя_пользователя>,<пароль>]

 Получение списка обнаруженных коллекций. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.collections.usage[строка_подкл,<имя_пользователя>,<пароль>]

 Получение статистики использования по коллекциям.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.connpool.stats[строка_подкл,<имя_пользователя>,<пароль>]

 Получение информации касательно открытых исходящих подключений с текущего экземпляра базы данных к другим членам кластера сегментов или набора реплик.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB;

mongodb.db.stats[строка_подкл,<имя_пользователя>,<пароль>,<база_данных>]

 Получение статистики, отражающей состояние системы заданной базы.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB;
- **база_данных** — имя базы (по умолчанию: admin).

mongodb.db.discovery[строка_подкл,<имя_пользователя>,<пароль>]

 Получение списка обнаруженных баз данных. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;

- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.jumbo_chunks.count[строка_подкл,<имя_пользователя>,<пароль>]

 Получение количества jumbo chunk.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.oplog.stats[строка_подкл,<имя_пользователя>,<пароль>]

 Получение состояния набора реплик, используя данные полученные от oplog.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.ping[строка_подкл,<имя_пользователя>,<пароль>]

 Проверка, можно ли установить подключение.
 Возвращаемое значение: *1* — подключение успешно; *0* — подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками).

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.rs.config[строка_подкл,<имя_пользователя>,<пароль>]

 Получение текущей конфигурации набора реплик.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.rs.status[строка_подкл,<имя_пользователя>,<пароль>]

 Получение состояния набора реплик с точки зрения члена, откуда вызывается метод.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.server.status[строка_подкл,<имя_пользователя>,<пароль>]

 Получение состояния базы данных.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.sh.discovery[строка_подкл,<имя_пользователя>,<пароль>]

 Получение списка обнаруженных сегментов, находящихся в кластере.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mongodb.version[строка_подкл,<имя_пользователя>,<пароль>]

 Получение версии сервера базы данных.
 Возвращаемое значение: *Строка*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные к MongoDB.

mqtt.get[<broker url>,topic,<user>,<password>]

 Подписывается на указанный topic или topics (с подстановочными символами) у заданного брокера и ожидает публикации.
 Возвращаемое значение: зависит от содержимого topic. Если используются подстановочные символы, возвращает содержимое topic в формате JSON.

Параметры:

- **broker url** - URL брокера MQTT в формате `protocol://host:port` без параметров запроса (поддерживаемые протоколы: `tcp`, `ssl`, `ws`). Если значение не указано, агент будет использовать `tcp://localhost:1883`. Если протокол или порт опущены, будут использованы протокол по умолчанию (`tcp`) или порт по умолчанию (1883);
- **topic** - topic MQTT (обязательно). Поддерживаются подстановочные символы (+,#);
- **user, password** - учетные данные для аутентификации (если требуется).

Комментарии:

- Элемент данных должен быть настроен как active check (тип элемента данных "Zabbix agent (active)");
- Сертификаты шифрования TLS можно использовать, сохранив их в расположение по умолчанию (например, каталог `/etc/ssl/certs/` для Ubuntu). Для TLS используйте схему `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Возвращает группы доступности.
 Возвращаемое значение: *JSON object*.

Параметры:

- **URI** - URI сервера MSSQL (единственная поддерживаемая схема — `sqlserver://`). Встроенные учетные данные будут проигнорированы. Можно указать имя экземпляра как часть URI, например: `sqlserver://localhost/InstanceName` (без порта). Если указан порт, имя экземпляра игнорируется;
- **user, password** - имя пользователя и пароль для отправки на защищенный сервер MSSQL.

Дополнительные сведения см. в файле readme MSSQL plugin.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *JSON object*.

Параметры:

- **URI** - URI сервера MSSQL (единственная поддерживаемая схема - `sqlserver://`). Встроенные учетные данные будут проигнорированы. Можно указать имя экземпляра как часть URI, например: `sqlserver://localhost/InstanceName` (без порта). Если указан порт, имя экземпляра игнорируется;
- **user, password** - имя пользователя, пароль для отправки на защищенный сервер MSSQL;
- **queryName** - имя пользовательского запроса, настроенного в `Plugins.MSSQL.CustomQueriesDir`, без расширения .sql;
- **args** - один или несколько аргументов, разделенных запятыми, для передачи запросу.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.db.get

 Возвращает все доступные базы данных MSSQL.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.job.status.get

 Возвращает статус заданий.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.last.backup.get

 Возвращает время последнего резервного копирования для всех баз данных.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.local.db.get

 Возвращает базы данных, которые участвуют в группе доступности Always On и реплике (основной или вторичной) и расположены на сервере, к которому было установлено соединение.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.mirroring.get

 Возвращает информацию о зеркалировании.
 Возвращаемое значение: *JSON-объект*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.nonlocal.db.get

 Возвращает базы данных, которые участвуют в группе доступности Always On и реплике (основной или вторичной), расположенной на других серверах (база данных не является локальной для экземпляра SQL Server, к которому было установлено подключение).
 Возвращаемое значение: *JSON object*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.perfcounter.get

 Возвращает счетчики производительности.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле README плагина MSSQL.

mssql.ping

 Проверка доступности базы данных. Проверяет, правильно ли настроено подключение.
 Возвращаемое значение: *1* - доступен, *0* - недоступен.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.quorum.get

 Возвращает информацию о кворуме.
 Возвращаемое значение: *JSON object*.

Для получения дополнительной информации см. файл readme плагина MSSQL.

mssql.quorum.member.get

 Возвращает членов кворума.
 Возвращаемое значение: *объект JSON*.

Дополнительную информацию см. в файле README плагина MSSQL.

mssql.replica.get

 Возвращает реплики.
 Возвращаемое значение: *JSON object*.

Дополнительную информацию см. в файле readme плагина MSSQL.

mssql.version

 Возвращает версию MSSQL.
 Возвращаемое значение: *String*.

Для получения дополнительной информации см. файл readme плагина MSSQL.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - URI или имя сеанса;
- **user, password** - учетные данные для входа в MySQL;
- **queryName** - имя пользовательского запроса, должно совпадать с именем файла SQL без расширения;
- **args** - один или несколько аргументов, разделенных запятыми, для передачи в запрос.

Дополнительные сведения см. в файле readme плагина MySQL, раздел *Custom queries*.

mysql.db.discovery[строка_подкл, <имя_пользователя>, <пароль>]

 Возвращает список баз данных MySQL. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: Результат SQL запроса «show databases» в формате LLD JSON.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL.

mysql.db.size[строка_подкл, <имя_пользователя>, <пароль>,<имя_базы>]

 Размер базы данных в байтах.
 Возвращаемое значение: Результат SQL запроса «select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?» для указанной базы данных в байтах.

Параметры:

- **строка_подкл** — URI или имя сессии;

- **имя_пользователя, пароль** — учётные данные для входа в MySQL;
- **имя_базы** — имя базы данных.

mysql.get_status_variables[строка_подкл, <имя_пользователя>, <пароль>]

 Значения глобальных переменных состояния.
 Возвращаемое значение: Результат SQL запроса «show global status» в формате JSON.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL.

mysql.ping[строка_подкл, <имя_пользователя>, <пароль>]

 Проверка, можно ли установить подключение.
 Возвращаемое значение: *1* — подключение успешно; *0* — подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками).

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL.

mysql.replication.discovery[строка_подкл, <имя_пользователя>, <пароль>]

 Возвращает список репликаций MySQL. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: результат SQL запроса `SHOW SLAVE STATUS` или `SHOW REPLICA STATUS` в формате LLD JSON.

Parameters:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL.

mysql.replication.get_slave_status[строка_подкл, <имя_пользователя>, <пароль>, <хост_мастера>]

 Состояние репликации.
 Возвращаемое значение: Результат SQL запроса SHOW SLAVE STATUS SHOW REPLICA STATUS' в формате JSON.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL;
- **хост_мастера** — Имя хоста мастера репликации. Если не найдено, возвращается ошибка. Если данный параметр не указан, возвращаются все хосты.

mysql.version[строка_подкл, <имя_пользователя>, <пароль>]

 Версия MySQL.
 Возвращаемое значение: *Строка* (с версией экземпляра MySQL).

Параметры:

- **строка_подкл** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные для входа в MySQL.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Выполняет DNS-запрос и возвращает подробную информацию о DNS-записи.
 Этот элемент данных является расширенной версией элемента данных `net.dns.record` агента Zabbix с поддержкой большего числа типов записей и настраиваемых флагов.
 Возвращаемые значения: *JSON object*

Параметры:

- **ip** - IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** - DNS-имя для запроса;
- **type** - тип записи, который нужно запросить (по умолчанию *SOA*);
- **timeout** - время ожидания запроса в секундах (по умолчанию 1 секунда);
- **count** - количество попыток выполнения запроса (по умолчанию 2);
- **protocol** - протокол, используемый для выполнения DNS-запросов: *udp* (по умолчанию) или *tcp*;
- **flags** - один или несколько аргументов, разделенных запятыми, которые передаются запросу.

Комментарии:

- Возможные значения `type`: *A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*, *TALINK*, *CDS*,

CDNSKEY, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY*. Обратите внимание, что значения должны быть только в верхнем регистре; значения в нижнем регистре или смешанном регистре не поддерживаются.

- Для обратного DNS-поиска (когда `type` установлен в *PTR*) можно указать DNS-имя как в обратном, так и в обычном формате (см. примеры ниже). Обратите внимание, что при запросе записи PTR DNS-имя фактически является IP-адресом.
- Возможные значения `flags`:
  - *cdflag* или *nocdflag* (по умолчанию);
  - *rdflag* (по умолчанию) или *nordflag*;
  - *dnssec* или *nodnssec* (по умолчанию);
  - *nsid* или *nonsid* (по умолчанию);
  - *edns0* (по умолчанию) или *noedns0*;
  - *aaflag* или *noaaflag* (по умолчанию);
  - *adflag* или *noadflag* (по умолчанию). Флаги `dnssec` и `nsid` нельзя использовать вместе с *noedns0*, так как для обоих требуется *edns0*. Обратите внимание, что значения должны быть только в нижнем регистре; значения в верхнем регистре или смешанном регистре не поддерживаются.
- Имена доменов с интернационализацией не поддерживаются, используйте вместо них имена, закодированные в IDNA.
- Вывод представляет собой объект, содержащий информацию о DNS-записи на основе указанных параметров (см. подробнее).

Примеры:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 Количество GPU-устройств.
 Возвращаемое значение: *Целое число*.

Дополнительную информацию смотрите в файле readme плагина NVIDIA GPU.

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

 Использование декодера устройства GPU в процентах.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Информация о режиме ECC GPU-устройства (текущий, ожидающий).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Статистика кодировщика устройства GPU.
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Использование кодировщика устройства GPU в процентах.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Общее энергопотребление GPU-устройства в миллиджоулях (mJ) с момента последней перезагрузки драйвера.
 Возвращаемое значение: *Целое число*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Информация об ошибках ECC памяти GPU-устройства (исправленные, неисправленные).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Информация об ошибках ECC регистра устройства GPU (исправленных, неисправленных).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Средняя скорость вентилятора GPU-устройства в процентах от максимальной скорости.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

nvml.device.get

 Обнаруженные GPU с UUID и именем.
 Возвращаемое значение: *JSON*.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Частота графического ядра GPU в МГц.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Дополнительную информацию см. в readme плагина NVIDIA GPU.

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

 Статистика памяти framebuffer устройства GPU (всего, зарезервировано, свободно, использовано).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Статистика памяти BAR1 устройства GPU (всего, свободно, использовано).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Частота памяти устройства GPU в МГц.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Использование PCI устройства GPU (пропускная способность передачи/приёма в KBps).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Состояние производительности устройства GPU.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

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

 Ограничение мощности GPU-устройства в милливаттах.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Текущее энергопотребление GPU-устройства в милливаттах.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

nvml.device.serial[<deviceUUID>]

 Серийный номер устройства GPU.
 Возвращаемое значение: *String*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Дополнительную информацию смотрите в readme плагина NVIDIA GPU.

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

 Тактовая частота потокового мультипроцессора устройства GPU в МГц.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

nvml.device.temperature[<deviceUUID>]

 Температура GPU-устройства в градусах Цельсия.
 Возвращаемое значение: *Integer*.

Параметр:

- **deviceUUID** — UUID GPU-устройства.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

nvml.device.utilization[<deviceUUID>]

 Статистика использования устройства GPU (использование GPU/памяти в процентах).
 Возвращаемое значение: *JSON*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

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

 Частота видеоядра GPU в МГц.
 Возвращаемое значение: *Целое число*.

Параметр:

- **deviceUUID** — UUID устройства GPU.

Для получения дополнительной информации см. readme плагина NVIDIA GPU.

nvml.system.driver.version

 Версия драйвера NVIDIA.
 Возвращаемое значение: *String*.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

nvml.version

 Версия библиотеки NVML.
 Возвращаемое значение: *String*.

Для получения дополнительной информации смотрите readme плагина NVIDIA GPU.

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Возвращает статистику групп дисков Automatic Storage Management (ASM).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **diskgroup** - имя группы дисков ASM, для которой выполняется запрос.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Возвращает список групп дисков ASM. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден, и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае значение будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.

- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

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

 Статистика архивных журналов.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **destination** - имя назначения для запроса.

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

 Информация о контейнерных базах данных (CDB).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и значение все равно будет обработано как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **database** - имя базы данных, к которой выполняется запрос.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Результат пользовательского запроса.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` задано значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;

- **service** - имя сервиса Oracle;
- **queryName** - имя пользовательского запроса, должно совпадать с именем SQL-файла без расширения;
- **args** - один или несколько аргументов, разделенных запятыми, которые передаются в запрос.

Комментарии: - Возвращаемые данные автоматически преобразуются в JSON. - Не возвращайте JSON напрямую из запросов, так как он будет поврежден, когда плагин попытается преобразовать его в JSON еще раз.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Возвращает статистику файлов данных.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из условий выше не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Возвращает список баз данных. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден, а `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Возвращает статистику Fast Recovery Area (FRA).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться и в любом случае будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

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

 Статистика экземпляра.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

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

 Информация о подключаемых базах данных (PDB).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет обработан как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **database** - имя базы данных для запроса.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Возвращает список PDB. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из условий выше не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Возвращает статистику Program Global Area (PGA).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:

    - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться и в любом случае будет трактоваться как URI;
    - имя сеанса - если такое имя найдено в файле конфигурации плагина;
    - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

oracle.ping[connString,<user>,<password>,<service>]

 Проверяет, можно ли установить соединение с Oracle.
 Возвращаемое значение: *1* - соединение успешно; *0* - соединение разорвано (если возникает любая ошибка, включая проблемы AUTH и конфигурации).

Параметры:

- **connString** - может быть:
    - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` задано значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и значение все равно будет обработано как URI;
    - имя сеанса - если такое имя найдено в файле конфигурации плагина;
    - ключ TNS - если для параметра ResolveTNS задано значение true и ни одно из указанных выше условий не выполняется;
    - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

Примеры:

```
oracle.ping[tcp://127.0.0.1:1521,ZABBIX_MON,zabbix,xe]
oracle.ping[localhost,ZABBIX_MON,zabbix,xe]
oracle.ping[zbx_tns_example,ZABBIX_MON,zabbix,xe]
oracle.ping["(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=x
```

oracle.proc.stats[connString,<user>,<password>,<service>]

 Возвращает статистику процессов.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
    - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
    - имя сеанса - если такое имя найдено в файле конфигурации плагина;
    - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
    - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

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

 Информация о файле журнала из файла управления.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с заданным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, ”tcp://”), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки ”(” (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Возвращает статистику System Global Area (SGA).
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с заданным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, ”tcp://”), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки ”(” (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Возвращает статистику сеансов.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, ”tcp://”), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться и в любом случае будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки ”(” (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **lockMaxTime** - максимальная длительность блокировки сеанса в секундах, при которой сеанс считается длительно заблокированным. По умолчанию: 600 секунд.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Возвращает набор значений системных метрик.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** - может иметь следующие значения:
  - URI - если сеанс с указанным именем не найден, а параметр `Plugins.Oracle.ResolveTNS` установлен в значение false. Если URI содержит схему (например, ”tcp://”), порт (например, 1521) или и то и другое, параметр ResolveTNS не учитывается, и значение в любом случае обрабатывается как URI;

  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle; поддерживается добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не зависит от регистра и не должен содержать завершающий пробел).
- **password** - пароль Oracle;
- **service** - имя службы Oracle;
- **duration** - интервал сбора значений системных метрик (в секундах). Возможные значения: *60* - длительный интервал (по умолчанию), *15* - короткий интервал.

oracle.sys.params[connString,<user>,<password>,<service>]

 Возвращает набор значений системных параметров.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - это может быть:
  - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` имеет значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS имеет значение true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

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

 Возвращает статистику по табличным пространствам.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае значение будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из условий выше не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **tablespace** - имя табличного пространства, для которого выполняется запрос. По умолчанию (если не указано, а `type` задан):
  - "TEMP" (если `type` задан как "TEMPORARY");
  - "USERS" (если `type` задан как "PERMANENT").
- **type** - тип табличного пространства, для которого выполняется запрос. По умолчанию (если `tablespace` задан): "PERMANENT".
- **conname** - имя контейнера, для которого требуется информация.

Если `tablespace`, `type` или `conname` не указаны, элемент данных вернет статистику по табличным пространствам для всех совпадающих контейнеров (включая PDB и CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Возвращает список табличных пространств. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с заданным именем не найден, а `Plugins.Oracle.ResolveTNS` установлен в false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться и в любом случае будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

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

 Возвращает информацию о пользователе Oracle.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и для `Plugins.Oracle.ResolveTNS` установлено значение false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle;
- **username** - имя пользователя, для которого нужна информация. Имена пользователей в нижнем регистре не поддерживаются. По умолчанию: текущий пользователь.

oracle.version[connString,<user>,<password>,<service>]

 Возвращает версию сервера базы данных.
 Возвращаемое значение: *String*.

Параметры:

- **connString** - может быть:
  - URI - если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлен в false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то и другое, параметр ResolveTNS не будет учитываться, и в любом случае он будет трактоваться как URI;
  - имя сеанса - если такое имя найдено в файле конфигурации плагина;
  - ключ TNS - если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значение TNS - если оно начинается с открывающей скобки "(" (начальные пробелы игнорируются). В этом случае параметр ResolveTNS не учитывается.
- **user** - имя пользователя Oracle, поддерживает добавление одного из параметров входа `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` или `as sysrac` в формате `user as sysdba` (параметр входа не чувствителен к регистру, не должен содержать пробел в конце).
- **password** - пароль Oracle;
- **service** - имя службы Oracle.

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

 Количество рабочих процессов autovacuum.
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;

- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Информация об архивированных файлах.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Общее количество контрольных точек для кластера базы данных, разбитое по типу контрольной точки.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Коэффициент попаданий в буферный кэш PostgreSQL.
 Возвращаемое значение: *Float*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Возвращает подключения по типу.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение Plugins.PostgreSQL.Default.Database из postgresql.conf);
- **queryName** - имя пользовательского запроса, должно совпадать с именем SQL-файла без расширения;
- **args** - один или несколько аргументов, разделенных запятыми, для передачи в запрос.

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

 Возраст самого старого FrozenXID в базе данных (число транзакций).
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Количество bloating-таблиц в каждой базе данных.
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Список баз данных PostgreSQL. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Размер базы данных в байтах.
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Собирает статистику по каждой базе данных. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Суммарные данные для всех баз данных в кластере.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Информация о предоставленных блокировках по каждой базе данных. Используется для обнаружения на низком уровне.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Возраст самого старого XID (количество транзакций).
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Проверяет, активно ли соединение.
 Возвращаемое значение: *1* - соединение активно; *0* - соединение разорвано (если возникает какая-либо ошибка, включая ошибки AUTH и конфигурации).

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Запрашивает метрики по времени выполнения.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`);
- **time period** - предел времени выполнения для подсчета медленных запросов (должно быть положительным целым числом).

pgsql.replication.count[uri,<имя_пользователя>,<пароль>]

 Количество серверов в режиме ожидания.
 Возвращаемое значение: *Целое число*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication.process[uri,<имя_пользователя>,<пароль>]

 Отставание сброса, отставание записи и отставание replay по каждому процессу sender.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication.process.discovery[uri,<имя_пользователя>,<пароль>]

 Обнаружение имени процесса репликации.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication.recovery_role[uri,<имя_пользователя>,<пароль>]

 Состояние восстановления.
 Возвращаемое значение: *0* — режим ведущего; *1* — восстановление ещё в процессе (режим ведомого).

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication.status[uri,<имя_пользователя>,<пароль>]

 Состояние репликации.
 Возвращаемое значение: *0* — передача потока не работает; *1* — передача потока работает; *2* — режим ведущего.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication_lag.b[uri,<имя_пользователя>,<пароль>]

 Отставание репликации в байтах.
 Возвращаемое значение: *Целое число*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

pgsql.replication_lag.sec[uri,<имя_пользователя>,<пароль>]

 Отставание репликации в секундах.
 Возвращаемое значение: *Целое число*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL.

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

 Время работы PostgreSQL в миллисекундах.
 Возвращаемое значение: *Float*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Возвращает версию PostgreSQL.
 Возвращаемое значение: *String*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

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

 Статистика WAL.
 Возвращаемое значение: *JSON object*.

Параметры:

- **uri** - URI или имя сеанса;
- **username, password** - учетные данные PostgreSQL;
- **database name** - имя базы данных (если не указано, используется значение `Plugins.PostgreSQL.Default.Database` из `postgresql.conf`).

redis.config[строка_подкл,<пароль>,<шаблон>]

 Получение параметров конфигурации экземпляра Redis, которые соответствуют шаблону.
 Возвращаемое значение: *JSON* — если использовался шаблон в стиле glob; одно значение — если шаблон не содержит какого-либо символа подстановки.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пароль** — пароль к Redis;
- **шаблон** — шаблон в стиле glob (по умолчанию *).

Комментарий:

- Предупреждение о безопасности: `redis.config` выполняет команду Redis `CONFIG GET` и может возвращать конфиденциальные параметры конфигурации (например, `requirepass`, `masterauth`, настройки, связанные с TLS, и другие секреты). Использование шаблона по умолчанию (`*`) или широких шаблонов может раскрыть эти секреты любому, кто в Zabbix имеет разрешение на просмотр значений элементов данных.

redis.info[строка_подкл,<пароль>,<раздел>]

 Получение вывода команды INFO.
 Возвращаемое значение: *JSON* — вывод преобразован в JSON.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пароль** — пароль к Redis;
- **раздел** — раздел [en] информации (по умолчанию *default*).

redis.ping[строка_подкл,<пароль>]

 Проверка, можно ли установить подключение.
 Возвращаемое значение: *1* — подключение успешно; *0* — подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками).

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пароль** — пароль к Redis.

redis.slowlog.count[строка_подкл,<пароль>]

 Количество медленных записей в журнале с момента запуска Redis.
 Возвращаемое значение: *Целое число*.

Параметры:

- **строка_подкл** — URI или имя сессии;
- **пароль** — пароль к Redis.

smart.attribute.discovery

 Возвращает список атрибутов устройств S.M.A.R.T.
 Возвращаемое значение: *JSON object*.

Комментарии:

- Возвращаются следующие макросы и их значения: `{#NAME}`, `{#DISKTYPE}`, `{#ID}`, `{#ATTRNAME}`, `{#THRESH}`;
- Поддерживаются типы накопителей HDD, SSD и NVME. Накопители могут использоваться отдельно или в составе RAID. У `{#NAME}` будет добавлено дополнение в случае RAID, например: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.discovery[<type>]

 Возвращает список устройств S.M.A.R.T.
 Возвращаемое значение: *JSON object*.

Параметр:

- **type** - задает значение, по которому выполняется сканирование дисков. Возможные значения: *id*, *name* (по умолчанию). *id* не поддерживается в Windows и при выборе вернет ошибку

Комментарии:

- Возвращаются следующие макросы и их значения: `{#NAME}`, `{#DISKTYPE}`, `{#MODEL}`, `{#SN}`, `{#PATH}`, `{#ATTRIBUTES}`, `{#RAIDTYPE}`;
- Поддерживаются типы накопителей HDD, SSD и NVME. Если накопитель не относится к RAID, `{#RAIDTYPE}` будет пустым. У `{#NAME}` будет добавлено дополнение в случае RAID, например: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.get[<path>,<raid type>]

 Возвращает все доступные свойства устройств S.M.A.R.T.
 Возвращаемое значение: *JSON object*.

Параметры:

- **path** - путь к диску, в качестве значения можно использовать макрос `{#PATH}`;
- **raid_type** - тип RAID, в качестве значения можно использовать макрос `{#RAID}`

Комментарии:

- Поддерживаются типы накопителей HDD, SSD и NVME. Накопители могут быть отдельными или объединенными в RAID;
- Данные включают версию smartctl и аргументы вызова, а также дополнительные поля:
disk_name - содержит имя с необходимыми дополнениями для обнаружения RAID, например: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - содержит тип диска HDD, SSD или NVME, например: `{"disk_type": "ssd"}`;
- Если параметры не указаны, элемент данных вернет информацию обо всех дисках.

systemd.unit.get[имя узла,<interface>]

 Возвращает все свойства systemd unit.
 Возвращаемое значение: *JSON object*.

Параметры:

- **имя узла** - имя unit (при необходимости можно использовать макрос `{#UNIT.NAME}` в прототипе элемента данных, чтобы определить имя);
- **interface** - тип интерфейса unit, возможные значения: *Unit* (по умолчанию), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Комментарии:

- Этот элемент данных поддерживается только на платформе Linux;
- LoadState, ActiveState и UnitFileState для интерфейса Unit возвращаются как текст и целое число: `"ActiveState":{"state":1`

systemd.unit.info[имя модуля,<свойство>,<интерфейс>]

 Информация о модуле systemd.
 Возвращаемое значение: *String*.

Параметры:

- **имя модуля** — имя модуля (для определения имени можно использовать макрос `{#UNIT.NAME}` в прототипе элемента данных);
- **свойство** — свойство модуля (например, ActiveState (по умолчанию), LoadState, Description);
- **интерфейс** — тип интерфейса модуля (например, Unit (по умолчанию), Socket, Service).

Примечания:

- Этот элемент данных поддерживается только на платформе Linux;
- Этот элемент данных позволяет получить определенное свойство интерфейса определенного типа, как описано в API D-Bus.

Примеры:

```
systemd.unit.info["{#UNIT.NAME}"] #                          (active, reloading, inactive, failed, activating
systemd.unit.info["{#UNIT.NAME}",LoadState] #                                 systemd
systemd.unit.info[mysqld.service,Id] #                        (mysqld.service)
systemd.unit.info[mysqld.service,Description] #                  (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #                                          (1562565036
systemd.unit.info[dbus.socket,NConnections,Socket] #
```

systemd.unit.discovery[<тип>]

 Список модулей systemd и их подробностей. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметр:

- **тип** — возможные значения: *all*, *automount*, *device*, *mount*, *path*, *service* (по умолчанию), *socket*, *swap*, *target*.

Этот элемент данных поддерживается только на платформе Linux.

web.certificate.get[hostname,<port>,<address>]

 Проверяет сертификаты и возвращает сведения о сертификате.
 Возвращаемое значение: *JSON object*.

Параметры:

- **hostname** - может быть IP-адресом или DNS-именем.
Может содержать схему URL (*только https*), путь (он будет проигнорирован) и порт.
Если порт указан и в первом, и во втором параметре, их значения должны совпадать.
Если указан address (3-й параметр), hostname используется только для SNI и проверки hostname;
- **port** - номер порта (по умолчанию для HTTPS используется 443);
- **address** - может быть IP-адресом или DNS-именем. Если указан, он будет использоваться для соединения, а hostname (1-й параметр) - для SNI и проверки host. Если 1-й параметр является IP-адресом, а 3-й параметр - DNS-именем, 1-й параметр будет использоваться для соединения, а 3-й параметр - для SNI и проверки host.

Комментарии:

- Этот элемент данных станет неподдерживаемым, если целевой адрес, указанный в конфигурации узла сети, не существует, недоступен или если TLS-handshake завершается с любой ошибкой, кроме недействительного сертификата;
- В настоящее время расширение X.509 AIA (Authority Information Access), CRL и OCSP (включая OCSP stapling), а также Certificate Transparency не поддерживаются;
- Поля JSON-ответа:
  - *x509*: содержит сведения о сертификате X.509.
    - *version*: версия X.509 (например, 3).
    - *serial_number*: серийный номер сертификата.
    - *signature_algorithm*: алгоритм, использованный для подписи сертификата (например, `SHA256-RSA`).
    - *issuer*: издатель сертификата.
    - *not_before*: дата начала действия сертификата.
    - *not_after*: дата окончания действия сертификата.
    - *subject*: субъект сертификата.
    - *public_key_algorithm*: алгоритм, использованный для открытого ключа (например, `RSA`).

        * *alternative_names*: альтернативные имена субъекта (если есть), иначе `null`.
    - result: содержит результат проверки.
        * *value*: статус проверки (см. возможные значения ниже).
        * *message*: подробное сообщение о проверке (например, `"certificate verified successfully"`).
    - *sha1_fingerprint*: SHA-1 отпечаток сертификата.
    - *sha256_fingerprint*: SHA-256 отпечаток сертификата.
- Поле `$.result.value` указывает результат проверки сертификата. Возможные значения:
    - *valid* - сертификат действителен и доверен.
    - *valid-but-self-signed* - сертификат действителен, но самоподписан, то есть его субъект совпадает с издателем.
    - *invalid* - сертификат недействителен из-за такой проблемы, как истечение срока действия, неверное hostname или неизвестный центр подписи.

Пример:

```
web.certificate.get[example.com,443]
```

JSON Response:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


Агент Zabbix для Windows

Обзор

Элементы данных агента Zabbix для Windows представлены в двух списках:

- Общие элементы данных — ключи элементов данных, которые являются общими с агентом Zabbix для UNIX;
- Элементы данных, специфичные для Windows — ключи элементов данных, которые поддерживаются **только** в Windows.

Обратите внимание, что все ключи элементов данных, поддерживаемые агентом Zabbix для Windows, также поддерживаются новым поколением Zabbix агента 2. Смотрите дополнительные ключи элементов данных, которые можно использовать только с агентом 2.

Смотрите также: Минимальные разрешения для элементов данных Windows.

Общие элементы данных

В таблице ниже перечислены элементы данных агента Zabbix, которые поддерживаются в Windows и являются общими с агентом Zabbix для UNIX:

- Ключ элемента данных является ссылкой на полные сведения об элементе данных агента Zabbix для UNIX
- Включены комментарии к элементам данных, относящиеся к Windows

Item key	Description	Item group
log	Мониторинг файла журнала. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	Мониторинг журналов
log.count	Количество совпавших строк в контролируемом файле журнала. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	
logrt	Мониторинг ротируемого файла журнала. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	
logrt.count	Количество совпавших строк в контролируемом ротируемом файле журнала. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	
modbus.get	Чтение данных Modbus.	Modbus
net.dns	Проверяет, доступна ли служба DNS. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix agent 2.	Сеть
net.dns.perf	Проверяет производительность службы DNS. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix agent 2.	
net.dns.record	Выполняет DNS-запрос. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix agent 2.	
net.if.discovery	Список сетевых интерфейсов. В некоторых версиях Windows, например Server 2008, для поддержки не-ASCII символов в именах интерфейсов могут потребоваться установленные последние обновления.	
net.if.in	Статистика входящего трафика на сетевом интерфейсе. В Windows элемент данных получает значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики интерфейсов были введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, агент использует 32-битные счетчики. Поддерживаются многобайтовые имена интерфейсов в Windows. Описания сетевых интерфейсов в Windows можно получить с помощью элементов данных net.if.discovery или net.if.list.	
net.if.out	Статистика исходящего трафика на сетевом интерфейсе. В Windows элемент данных получает значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики интерфейсов были введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, агент использует 32-битные счетчики. Поддерживаются многобайтовые имена интерфейсов в Windows. Описания сетевых интерфейсов в Windows можно получить с помощью элементов данных net.if.discovery или net.if.list.	
net.if.total	Сумма статистики входящего и исходящего трафика на сетевом интерфейсе. В Windows элемент данных получает значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики интерфейсов были введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, агент использует 32-битные счетчики. Поддерживаются многобайтовые имена интерфейсов в Windows. Описания сетевых интерфейсов в Windows можно получить с помощью элементов данных net.if.discovery или net.if.list.	
net.tcp.listen	Проверяет, находится ли этот TCP-порт в состоянии LISTEN.	
net.tcp.port	Проверяет, возможно ли установить TCP-соединение с указанным портом.	
net.tcp.service	Проверяет, запущена ли служба и принимает ли она TCP-соединения. Проверка LDAP и HTTPS в Windows поддерживается только Zabbix agent 2.	
net.tcp.service.perf	Проверяет производительность TCP-службы. Проверка LDAP и HTTPS в Windows поддерживается только Zabbix agent 2.	

Item key	Description	Item group
net.tcp.socket.count	Возвращает количество TCP-сокетов, соответствующих параметрам. Этот элемент данных поддерживается в Linux агентом Zabbix, но в Windows он поддерживается только Zabbix agent 2 в 64-битной Windows.	
net.udp.service	Проверяет, запущена ли служба и отвечает ли она на UDP-запросы.	
net.udp.service.perf	Проверяет производительность UDP-службы.	
net.udp.socket.count	Возвращает количество UDP-сокетов, соответствующих параметрам. Этот элемент данных поддерживается в Linux агентом Zabbix, но в Windows он поддерживается только Zabbix agent 2 в 64-битной Windows.	
proc.get	Список процессов ОС и их параметров. Параметр `cmdline` не поддерживается в Windows.	Процессы
proc.num	Количество процессов. В Windows поддерживаются только параметры `name` и `user`.	
system.cpu.discovery	Список обнаруженных CPU/ядер CPU.	Система
system.cpu.load	Нагрузка CPU. Когда в агенте Zabbix запускается процесс сборщика, инициализируются следующие счетчики производительности, которые затем используются для этого элемента данных: `\System\Processor Queue Length`	
system.cpu.num	Количество CPU.	
system.cpu.util	Процент загрузки CPU. Значение получается с использованием счетчика производительности *Processor Time*. Обратите внимание, что начиная с Windows 8 Диспетчер задач показывает загрузку CPU на основе счетчика *Processor Utility*, тогда как в предыдущих версиях использовался счетчик *Processor Time* (см. подробности). На Windows поддерживается только параметр `type` со значением `system`.	
system.hostname	Имя узла сети в системе. Значение в Windows получается с помощью функций `GetComputerName()` (для **netbios**), `GetComputerNameExA()` (для **fqdn**) или `gethostname()` (для **host**). См. также более подробное описание.	
system.localtime	Системное время.	
system.run	Выполнить указанный команду на узле сети.	
system.sw.arch	Сведения об архитектуре программного обеспечения.	
system.sw.os	Сведения об операционной системе.	
system.sw.os.get	Подробные сведения об операционной системе (версия, тип, имя дистрибутива, минорная и мажорная версии и т. д.).	
system.swap.size	Размер пространства подкачки в байтах или в процентах от общего объема. Параметр типа `pused` поддерживается в Linux агентом Zabbix, но в Windows он поддерживается только Zabbix agent 2. Обратите внимание, что этот ключ может возвращать некорректный размер/процент пространства подкачки на виртуализированных платформах Windows (VMware ESXi, VirtualBox). В этом случае можно использовать ключ `perf_counter[\700(_Total)\702]`, чтобы получить корректный процент пространства подкачки.	
system.uname	Идентификация системы. В Windows значение этого элемента данных получается из классов WMI `Win32_OperatingSystem` и `Win32_Processor`. Имя ОС (включая редакцию) может быть переведено на язык отображения пользователя. В некоторых версиях Windows оно содержит символы товарных знаков и лишние пробелы.	
system.uptime	Время работы системы в секундах.	
vfs.dir.count	Количество записей каталога. В Windows символьные ссылки на каталоги пропускаются, а жесткие ссылки учитываются только один раз.	Виртуальные файловые системы
vfs.dir.get	Список записей каталога. В Windows символьные ссылки на каталоги пропускаются, а жесткие ссылки учитываются только один раз.	
vfs.dir.size	Размер каталога. В Windows любая символьная ссылка пропускается, а жесткие ссылки учитываются только один раз.	
vfs.file.cksum	Контрольная сумма файла, вычисленная по алгоритму UNIX cksum.	
vfs.file.contents	Получение содержимого файла.	

Item key	Description	Item group
vfs.file.exists	Проверяет, существует ли файл. В Windows двойные кавычки нужно экранировать обратным слешем \\, а весь ключ элемента данных заключать в двойные кавычки при использовании утилиты командной строки для вызова `zabbix_get.exe` или `agent2`. Обратите внимание, что элемент данных может перейти в состояние unsupported в Windows, если выполняется поиск каталога внутри несуществующего каталога, например `vfs.file.exists[C:\no\dir,dir]` (где 'no' не существует).	
vfs.file.get	Возвращает сведения о файле. Поддерживаемые типы файлов в Windows: обычный файл, каталог, символьная ссылка	
vfs.file.md5sum	MD5-контрольная сумма файла.	
vfs.file.owner	Получает владельца файла.	
vfs.file.regexp	Извлекает строку из файла.	
vfs.file.regmatch	Находит строку в файле.	
vfs.file.size	Размер файла.	
vfs.file.time	Сведения о времени файла. В Windows XP `vfs.file.time[file,change]` может быть равен `vfs.file.time[file,access]`.	
vfs.fs.discovery	Список смонтированных файловых систем с их типом и параметрами монтирования. Макрос {#FSLABEL} поддерживается в Windows.	
vfs.fs.get	Список смонтированных файловых систем с их типом, доступным дисковым пространством, статистикой inode и параметрами монтирования. Макрос {#FSLABEL} поддерживается в Windows.	
vfs.fs.size	Дисковое пространство в байтах или в процентах от общего объема.	
vm.memory.size	Размер памяти в байтах или в процентах от общего объема.	Виртуальная память
web.page.get	Получение содержимого веб-страницы.	Мониторинг веба
web.page.perf	Время загрузки полной веб-страницы.	
web.page.regexp	Поиск строки на веб-странице.	
agent.hostmetadata	Метаданные узла сети агента.	Zabbix
agent.hostname	Имя узла сети агента.	
agent.ping	Проверка доступности агента.	
agent.variant	Вариант агента Zabbix (Zabbix agent или Zabbix agent 2).	
agent.version	Версия агента Zabbix.	
zabbix.stats	Удаленно возвращает набор внутренних метрик сервера Zabbix или прокси.	
zabbix.stats (queue)	Удаленно возвращает количество контролируемых элементов данных в очереди, задержанных на сервере Zabbix или прокси.	

Элементы данных, специфичные для Windows

В таблице приведены сведения о ключах элементов данных, которые поддерживаются **только** Zabbix агентом для Windows.

Элементы данных, специфичные для Windows, иногда являются приблизительным аналогом аналогичного элемента данных агента; например, `proc_info`, поддерживаемый в Windows, примерно соответствует элементу данных `proc.mem`, который в Windows не поддерживается.

Ключ элемента данных является ссылкой на полную информацию о нём.

Ключ элемента данных	Описание	Группа элементов данных
eventlog	Мониторинг журналов событий Windows.	Log monitoring
eventlog.count	Подсчёт строк в журнале событий Windows.	
net.if.list	Список сетевых интерфейсов (включая тип, состояние, IPv4 адрес, описание интерфейса.	Network

Ключ элемента данных	Описание	Группа элементов данных
perf_counter	Значение любого счётчика производительности Windows.	Performance counters
perf_counter_en	Значение любого счётчика производительности Windows на английском языке.	
perf_instance.discovery	Список экземпляров объекта счётчиков производительности Windows.	
perf_instance_en.discovery	Список экземпляров объекта счётчиков производительности Windows, обнаруженных с использованием имён объектов на английском языке.	
proc_info	Различная информация об указанном процессе(-ах).	Processes
registry.data	Получение данных для указанного имени значения в ключе Реестра Windows.	Registry
registry.get	Список значений или ключей Реестра Windows, расположенных в указанном ключе.	
service.discovery	Список служб Windows.	Services
service.info	Информация о службе.	
services	Список служб.	
vm.vmemory.size	Размер виртуального пространства в байтах или в процентах от общего размера.	Virtual memory
wmi.get	Выполнение WMI запроса и получение первого выбранного объекта.	WMI
wmi.getall	Выполнение WMI запроса и получение полного ответа.	

Подробности ключей элементов данных

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками **< >**, опциональны.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 Мониторинг журнала событий.
 Возвращаемое значение: *Log*.

Параметры:

- **name** - имя канала журнала событий (*Log Name* в графическом интерфейсе Event Viewer);
- **regexp** - регулярное выражение, описывающее требуемый шаблон (с учетом регистра);
- **severity** - регулярное выражение, описывающее уровень серьезности (без учета регистра). Этот параметр принимает регулярное выражение на основе следующих значений: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (при работе в Windows Vista или более новой версии).
- **source** - регулярное выражение, описывающее идентификатор источника (без учета регистра);
- **eventid** - регулярное выражение, описывающее идентификатор(ы) события (с учетом регистра);
- **maxlines** - максимальное число новых строк в секунду, которое агент будет отправлять на сервер Zabbix или прокси. Этот параметр переопределяет значение `MaxLinesPerSecond` в `zabbix_agentd.conf`.
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропуск обработки более старых данных (влияет только на вновь созданные элементы данных).

Комментарии:

- Элемент данных должен быть настроен как активная проверка;
- Агент не может отправлять события из журнала ”Forwarded events”;
- Поддерживается Windows Eventing 6.0;
- Выбор для этого элемента данных не-Log типа информации приведет к потере локальной метки времени, а также сведений об уровне серьезности и источнике журнала;
- См. также дополнительную информацию о мониторинге журналов.

Примеры:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] #                                    `TWOSHORT` (            *Matches regu
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Количество строк в журнале событий Windows.
 Возвращаемое значение: *Integer*.

Параметры:

- **name** - имя канала журнала событий (*Log Name* в графическом интерфейсе Event Viewer);
- **regexp** - регулярное выражение, описывающее требуемый шаблон (с учетом регистра);
- **severity** - регулярное выражение, описывающее уровень серьезности (без учета регистра). Этот параметр принимает регулярное выражение на основе следующих значений: "Information", "Warning", "Error", "Critical", "Verbose" (при работе в Windows Vista или более новой версии).
- **source** - регулярное выражение, описывающее идентификатор источника (без учета регистра);
- **eventid** - регулярное выражение, описывающее идентификатор(ы) события (с учетом регистра);
- **maxproclines** - максимальное количество новых строк в секунду, которое агент будет анализировать (не может превышать 10000). Значение по умолчанию: 10*MaxLinesPerSecond в zabbix_agentd.conf.
- **mode** - возможные значения:
  - *all* (по умолчанию);
  - *skip* - пропускать обработку более старых данных (влияет только на вновь созданные элементы данных).

Комментарии:

- Элемент данных должен быть настроен как активная проверка;
- Агент не может отправлять события из журнала "Forwarded events";
- Поддерживается Windows Eventing 6.0;
- Выбор для этого элемента данных не-Log типа информации приведет к потере локальной метки времени, а также информации о серьезности и источнике журнала;
- См. также дополнительную информацию о мониторинге журналов.

Примеры:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 Список сетевых интерфейсов (включая тип, состояние, IPv4 адрес, описание интерфейса).
 Возвращаемое значение: *Текст*.

Комментарии:

- Поддерживаются мультибайтовые имена интерфейса;
- Отключённые интерфейсы не входят в список;
- Включение/отключение некоторых компонентов Windows может изменить порядок имён интерфейсов в Windows;
- В некоторых версиях Windows (к примеру, Server 2008) может потребоваться установка последних обновлений для поддержки не-ASCII символов в именах интерфейсов.

perf_counter[counter,<interval>]

 Значение любого счетчика производительности Windows.
 Возвращаемое значение: *Integer*, *float*, *string* или *text* (в зависимости от запроса).

Параметры:

- **counter** - путь к счетчику;
- **interval** - последние N секунд для хранения среднего значения. Значение `interval` должно быть от 1 до 900 секунд включительно, а значение по умолчанию — 1.

Комментарии:

- Этот элемент данных следует UNC-путям, что может представлять риск безопасности;
- `interval` используется для счетчиков, которым требуется более одного измерения (например, для загрузки CPU), поэтому проверка каждый раз возвращает среднее значение за последние "interval" секунд;
- Для получения списка доступных счетчиков можно использовать Performance Monitor.
- См. также: Счетчики производительности Windows.

perf_counter_en[counter,<interval>]

 Значение любого счетчика производительности Windows на английском языке.
 Возвращаемое значение: *Integer*, *float*, *string* или *text* (в зависимости от запроса).

Параметры:

- **counter** - путь к счетчику на английском языке;
- **interval** - последние N секунд для хранения среднего значения. `interval` должен быть от 1 до 900 секунд включительно, значение по умолчанию — 1.

Комментарии:

- Этот элемент данных следует по UNC-путям, что может представлять риск безопасности;
- `interval` используется для счетчиков, которым требуется более одного измерения (например, для загрузки CPU), поэтому проверка каждый раз возвращает среднее значение за последние "interval" секунд;
- Этот элемент данных поддерживается только в **Windows Server 2008/Vista** и более поздних версиях;
- Список английских строк можно найти, просмотрев следующий раздел реестра: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[объект]

 Список экземпляров объекта счётчиков производительности Windows. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметр:

- **объект** — имя объекта (локализованное).

perf_instance_en.discovery[объект]

 Список экземпляров объекта счётчиков производительности Windows, обнаруженных с использованием имён объектов на английском языке. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*.

Параметр:

- **объект** — имя объекта (на английском языке).

proc_info[process,<attribute>,<type>]

 Различная информация о конкретном процессе(ах).
 Возвращаемое значение: *Float*.

Параметры:

- **process** - имя процесса;
- **attribute** - запрашиваемый атрибут процесса;
- **type** - тип представления (имеет значение, когда существует более одного процесса с одинаковым именем)

Комментарии:

- Поддерживаются следующие `attributes`:
  - *vmsize* (по умолчанию) - размер виртуальной памяти процесса в Kbytes;
  - *wkset* - размер рабочего набора процесса (объем физической памяти, используемой процессом) в Kbytes;
  - *pf* - количество page faults;
  - *ktime* - время ядра процесса в миллисекундах;
  - *utime* - пользовательское время процесса в миллисекундах;
  - *io_read_b* - количество байтов, прочитанных процессом во время операций ввода-вывода;
  - *io_read_op* - количество операций чтения, выполненных процессом;
  - *io_write_b* - количество байтов, записанных процессом во время операций ввода-вывода;
  - *io_write_op* - количество операций записи, выполненных процессом;
  - *io_other_b* - количество байтов, переданных процессом во время операций, отличных от операций чтения и записи;
  - *io_other_op* - количество операций ввода-вывода, выполненных процессом, кроме операций чтения и записи;
  - *gdiobj* - количество объектов GDI, используемых процессом;
  - *userobj* - количество объектов USER, используемых процессом.
- Допустимые `types`:
  - *avg* (по умолчанию) - среднее значение для всех процессов с именем <process>;
  - *min* - минимальное значение среди всех процессов с именем <process>;
  - *max* - максимальное значение среди всех процессов с именем <process>;
  - *sum* - сумма значений для всех процессов с именем <process>.
- На 64-разрядной системе для корректной работы этого элемента данных требуется 64-разрядный агент Zabbix.

Примеры:

```
proc_info[iexplore.exe,wkset,sum] #                    ,                  Internet Explorer
proc_info[iexplore.exe,pf,avg] #                page faults          Internet Explorer
```

registry.data[key,<value name>]

 Возвращает данные для указанного имени значения в ключе реестра Windows.
 Возвращаемое значение: *Integer*, *string* или *text* (в зависимости от типа значения)

Параметры:

- **key** - ключ реестра, включая корневой ключ; допускаются сокращения корневых ключей (например, HKLM);

- **value name** - имя значения реестра в ключе (по умолчанию пустая строка ""). Если имя значения не указано, возвращается значение по умолчанию.

Комментарии:

- Поддерживаемые сокращения корневых ключей:
  - HKCR - HKEY_CLASSES_ROOT
  - HKCC - HKEY_CURRENT_CONFIG
  - HKCU - HKEY_CURRENT_USER
  - HKCULS - HKEY_CURRENT_USER_LOCAL_SETTINGS
  - HKLM - HKEY_LOCAL_MACHINE
  - HKPD - HKEY_PERFORMANCE_DATA
  - HKPN - HKEY_PERFORMANCE_NLSTEXT
  - HKPT - HKEY_PERFORMANCE_TEXT
  - HKU - HKEY_USERS
- Ключи с пробелами должны быть заключены в двойные кавычки.

Примеры:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #return the data of
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #return the data of t
```

registry.get[key,<mode>,<name regexp>]

 Список значений или ключей реестра Windows, расположенных по указанному ключу.
 Возвращаемое значение: *JSON object*.

Параметры:

- **key** - ключ реестра, включая корневой ключ; допускаются сокращения корневых ключей (например, HKLM) (см. комментарии к `registry.data[]` для полного списка сокращений);
- **mode** - возможные значения:
values (по умолчанию) или *keys*;
- **name regexp** - обнаруживать только значения, имена которых соответствуют regexp (по умолчанию - обнаруживать все значения). Разрешено только при *values* в качестве *mode*.

Ключи с пробелами должны быть заключены в двойные кавычки.

Примеры:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON                    ,           ,         ,                    .
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #
JSON                    ,           ,         ,                    .
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #                    .
JSON                                .
```

service.discovery

 Список служб Windows. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *объект JSON*.

service.info[service,<param>]

 Информация о службе.
 Возвращаемое значение: - *Integer* - при `param` = *state*, *startup*;

 Для *state*: - 0 - выполняется - 1 - приостановлена - 2 - ожидает запуска - 3 - ожидает приостановки - 4 - ожидает продолжения - 5 - ожидает остановки - 6 - остановлена - 7 - неизвестно - 255 - такой службы нет

 Для *startup*: - 0 - автоматический - 1 - автоматический с задержкой - 2 - вручную - 3 - отключена - 4 - неизвестно - 5 - автоматический запуск по триггеру - 6 - автоматический запуск по триггеру с задержкой - 7 - ручной запуск по триггеру - *String* - при `param` = *displayname*, *path*, *user*; - *Text* - при `param` = *description*.

Параметры:

- **service** - реальное имя службы или ее отображаемое имя, как оно указано в оснастке MMC Services;
- **param** - *state* (по умолчанию), *displayname*, *path*, *user*, *startup* или *description*.

Комментарии:

- Элементы данных `service.info[service,state]` и `service.info[service]` возвращают одну и ту же информацию;
- Только при `param` = *state* этот элемент данных возвращает значение для несуществующих служб (255).

Примеры:

```
service.info[SNMPTRAP] -           SNMPTRAP;
service.info[SNMP Trap] -               ,                    ;
service.info[EventLog,startup] -            EventLog
```

services[<тип>,<состояние>,<исключения>]

 Список служб.
 Возвращаемое значение: *0* — если список служб пуст; *Текст* — список служб, разделенных новой строкой.

Параметры:

- **тип** — *all* (по умолчанию), *automatic*, *manual* или *disabled*;
- **состояние** — *all* (по умолчанию), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* или *paused*;
- **исключения** — список служб, исключаемых из результата. Исключаемые службы должны быть указаны в двойных кавычках, разделены запятой, без пробелов.

Примеры:

```
services[,started] #                   ;
services[automatic, stopped] #                     ,                ;
services[automatic, stopped, "service1,service2,service3"] #                     ,                ,
```

vm.vmemory.size[<тип>]

 Размер виртуальной памяти в байтах или в процентах от общего размера.
 Возвращаемое значение: *Целое число* — для байт; *Число с плавающей точкой* — для процентов.

Параметр:

- **тип** — возможные значения: *available* (доступно виртуальной памяти), *pavailable* (доступно виртуальной памяти, в процентах), *pused* (использовано виртуальной памяти, в процентах), *total* (всего виртуальной памяти, по умолчанию) или *used* (использовано виртуальной памяти)

Комментарии:

- Мониторинг статистики виртуальной памяти основывается на:
  - Общем количестве виртуальной памяти в Windows (всего физической + размер файла подкачки);
  - Максимальном количестве памяти, которое может занять Zabbix агент;
  - Текущем лимите выделенной памяти системе или Zabbix агенту, смотря что меньше.

Пример:

```
vm.vmemory.size[pavailable] #                     ,
```

wmi.get[<пространство_имён>,<запрос>]

 Выполнение WMI запроса и получение первого выбранного объекта.
 Возвращаемое значение: *Целое число*, *число с плавающей точкой*, *строка* или *текст* (в зависимости от запроса).

Параметры:

- **пространство_имён** — название пространства имён WMI;
- **запрос** — WMI запрос, возвращающий один объект.

WMI запросы выполняются с использованием WQL [en].

Пример:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #
```

wmi.getall[<пространство_имён>,<запрос>]

 Выполнение WMI запроса и получение полного ответа. Можно использовать для низкоуровневого обнаружения.
 Возвращаемое значение: *Объект JSON*

Параметры:

- **пространство_имён** — название пространства имён WMI;
- **запрос** — WMI запрос.

Комментарии:

- Запросы WMI выполняются при помощи WQL [en].
- Можно использовать JSONPath предобработку при обращении к более конкретным значениям в полученном JSON.

Пример:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #
```

Мониторинг служб Windows

Это руководство содержит пошаговые инструкции по настройке мониторинга служб Windows. Предполагается, что Zabbix сервер и агент уже настроены и работают.

Шаг 1

Получите имя службы.

Вы можете узнать имя службы, открыв оснастку MMC Services и вызвав свойства службы. На вкладке *General* вы должны увидеть поле с названием *Service name*. Следующее за ним значение — это имя, которое вы будете использовать при настройке элемента данных для мониторинга. Например, если вы хотите отслеживать службу "workstation", то имя вашей службы может быть таким: **lanmanworkstation**.

Шаг 2

Настройте элемент данных для наблюдения за службой.

Элемент данных `service.info[ ,< >]` возвращает информацию о конкретной службе. В зависимости от требуемой вам информации, укажите опцию , которая принимает следующие значения: *displayname*, *state*, *path*, *user*, *startup* или *description*. Значением по умолчанию является *state*, если не указан (`service.info[ ]`).

Тип возвращаемого значения зависит от выбранного : целое число при *state* и *startup*; строка символов при *displayname*, *path* и *user*; текст при *description*.

Пример:

- *Ключ*: `service.info[lanmanworkstation]`
- *Тип информации*: Числовой (целое положительное)

Элемент данных `service.info[lanmanworkstation]` будет извлекать информацию о состоянии службы как числовое значение. Чтобы преобразовать числовое значение в текстовое представление в веб-интерфейсе («0» как «Running», «1» как «Paused» и т.д.), вы можете настроить преобразование значений на узле сети, на котором сконфигурирован элемент данных. Чтобы это сделать, либо присоедините шаблон *Windows services by Zabbix agent* или *Windows services by Zabbix agent active* к узлу сети, либо настройте на узле сети новое преобразование значений, основанное на преобразованиях значений *Windows service state*, уже настроенных в упомянутых шаблонах.

Обратите внимание, что оба упомянутых шаблона имеют правило обнаружения, настроенное для автоматического обнаружения служб. Если вы не хотите этого, вы можете деактивировать правило обнаружения на уровне узла сети после присоединения к нему шаблона.

Обнаружение служб Windows

Низкоуровневое обнаружение даёт возможность автоматического создания элементов данных, триггеров и графиков по различных объектам на компьютере. Zabbix может автоматически начать наблюдение за службами Windows на вашей машине, без необходимости знания точного имени службы или создания элементов данных по каждой службе вручную. Можно использовать фильтр для генерирования реальных элементов данных, триггеров и графиков только по интересующим службам.

Мониторинг файла журнала

Обзор

Zabbix можно использовать для централизованного мониторинга и анализа файлов журналов с поддержкой или без поддержки ротации журналов.

Уведомления можно использовать, чтобы предупреждать пользователей, когда файл журнала содержит определенные строки или шаблоны строк.

Чтобы отслеживать файл журнала, у вас должны быть:

- Запущен агент Zabbix на узле сети.
- Настроен элемент данных для мониторинга журнала.

Attention:
Ограничение размера отслеживаемого файла журнала зависит от поддержки больших файлов.

Конфигурация

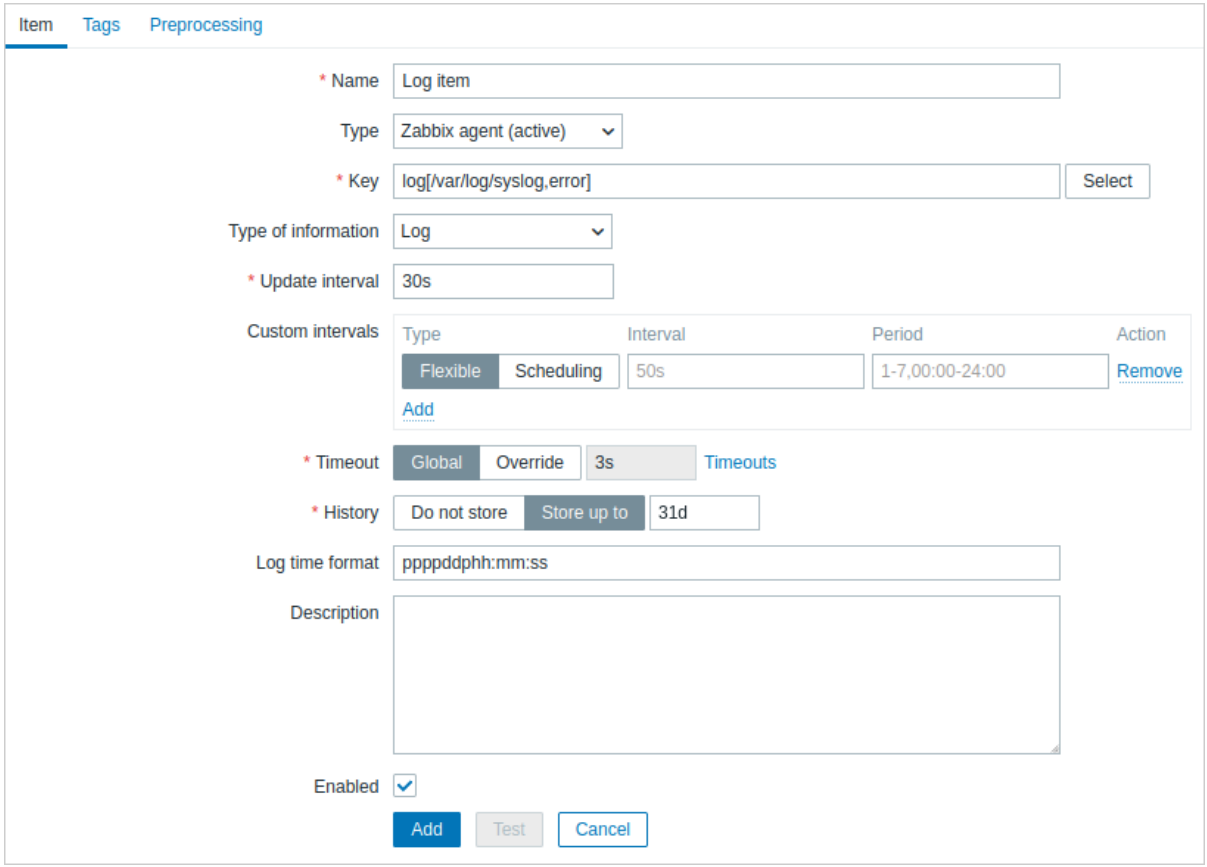
Проверка параметров агента

Убедитесь, что в файле конфигурации агента:

- параметр `Hostname` совпадает с именем узла сети во веб-интерфейсе.
- в параметре `ServerActive` указаны серверы для обработки активных проверок.

Настройка элемента данных

Настройте элемент данных для мониторинга журнала.



Все обязательные поля ввода отмечены красной звездочкой.

Для элементов данных мониторинга журналов необходимо указать:

Тип	Выберите здесь **Zabbix agent (active)**.
Ключ	Используйте один из следующих ключей элементов данных: `log[]` или `logrt[]`: Эти два ключа элементов данных позволяют отслеживать журналы и фильтровать записи журнала по регулярному выражению содержимого, если оно указано. Например: `log[/var/log/syslog,error]`. Убедитесь, что пользователь 'zabbix' имеет разрешения на чтение файла; в противном случае статус элемента данных будет установлен в значение «не поддерживается». **log.count[]** или **logrt.count[]**: Эти два ключа элементов данных позволяют возвращать только количество соответствующих строк. Подробнее об использовании этих ключей элементов данных и их параметров см. в разделе о поддерживаемых ключах элементов данных агента Zabbix.
Тип информации	Заполняется автоматически: для элементов данных `log[]` или `logrt[]` - *Журнал*; для элементов данных `log.count[]` или `logrt.count[]` - *Числовой (без знака)*. При использовании параметра `output` можно вручную выбрать подходящий тип информации, отличный от *Журнал*. Обратите внимание, что выбор типа информации, отличного от «Журнал», приведет к потере локальной временной метки.

Интервал обновления (в секундах)	Параметр определяет, как часто агент Zabbix будет проверять наличие изменений в файле журнала. Установка значения 1 секунда гарантирует получение новых записей как можно скорее.
Формат времени журнала	В этом поле можно указать шаблон для разбора временной метки в строке журнала. Поддерживаются следующие обозначения: * **y**: *Год (1970-2038)* * **M**: *Месяц (01-12)* * **d**: *День (01-31)* * **h**: *Час (00-23)* * **m**: *Минута (00-59)* * **s**: *Секунда (00-59)* Если оставить поле пустым, временная метка будет установлена в значение 0 в формате времени Unix, что соответствует `January 1, 1970`. Например, рассмотрим следующую строку из файла журнала агента Zabbix: `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` Она начинается с шести позиций символов для PID, за которыми следуют дата, время и остальная часть сообщения. Формат времени журнала для этой строки будет иметь вид ”ppppp:yyyyMMdd:hhmmss”. Обратите внимание, что символы ”p” и ”:” являются заполнителями и могут быть любыми символами, кроме ”yMdhms”.

Важные примечания

- Сервер и агент ведут учет размера отслеживаемого журнала и времени его последнего изменения (для `logrt`) в двух счетчиках. Кроме того:
  - Агент также внутренне использует номера inode (в UNIX/GNU/Linux), индексы файлов (в Microsoft Windows) и MD5-суммы первых 512 байт файла журнала для улучшения решений, когда файлы журналов усекаются и ротируются.
  - В системах UNIX/GNU/Linux предполагается, что файловые системы, где хранятся файлы журналов, сообщают номера inode, которые можно использовать для отслеживания файлов.
  - В Microsoft Windows агент Zabbix определяет тип файловой системы, на которой находятся файлы журналов, и использует:
    * В файловых системах NTFS 64-битные индексы файлов.
    * В файловых системах ReFS (только начиная с Microsoft Windows Server 2012) 128-битные идентификаторы файлов.
    * В файловых системах, где индексы файлов изменяются (например, FAT32, exFAT), используется резервный алгоритм, чтобы принимать разумное решение в неопределенных условиях, когда ротация файла журнала приводит к нескольким файлам журнала с одинаковым временем последнего изменения.
  - Номера inode, индексы файлов и MD5-суммы внутренне собираются агентом Zabbix. Они не передаются на сервер Zabbix и теряются при остановке агента Zabbix.
  - Не изменяйте время последнего изменения файла журнала (например, с помощью `touch`) и не заменяйте отслеживаемый файл журнала путем копирования файла обратно под его исходное имя (это создает новый inode). В любом из этих случаев Zabbix может считать файл другим файлом и перечитать его с начала, что может привести к дублирующимся оповещениям.
  - Если для элемента данных `logrt[]` есть несколько совпадающих файлов и агент Zabbix отслеживает самый последний из них, а этот самый последний файл журнала удаляется, в журнале записывается предупреждение `"there are no files matching "<regexp mask>" in "<directory>"`. Агент Zabbix игнорирует файлы журналов со временем изменения меньше, чем самое позднее время изменения, увиденное агентом для проверяемого элемента данных `logrt[]`.
- Агент начинает читать файл журнала с того места, на котором остановился в прошлый раз.
- Количество уже проанализированных байт (счетчик размера) и время последнего изменения (счетчик времени) хранятся в базе данных Zabbix и передаются агенту, чтобы гарантировать, что агент начнет чтение файла журнала с этого места в случаях, когда агент только что запущен или получил элементы данных, которые ранее были отключены или не поддерживались. Однако если агент получил от сервера ненулевой счетчик размера, но элемент данных `logrt[]` или `logrt.count[]` не может найти совпадающие файлы, счетчик размера сбрасывается в 0, чтобы анализировать с начала, если файлы появятся позже.
- Каждый раз, когда файл журнала становится меньше, чем известный агенту счетчик размера журнала, счетчик сбрасывается в ноль, и агент начинает читать файл журнала с начала, принимая во внимание счетчик времени.
- Если в каталоге есть несколько совпадающих файлов с одинаковым временем последнего изменения, агент пытается корректно проанализировать все файлы журнала с одинаковым временем изменения и избежать пропуска данных или повторного анализа одних и тех же данных, хотя это не может быть гарантировано во всех

ситуациях. Агент не предполагает какой-либо конкретной схемы ротации файлов журнала и не определяет ее. При наличии нескольких файлов журнала с одинаковым временем последнего изменения агент будет обрабатывать их в лексикографически убывающем порядке. Таким образом, для некоторых схем ротации файлы журнала будут анализироваться и сообщаться в их исходном порядке. Для других схем ротации исходный порядок файлов журнала не будет соблюдаться, что может привести к сообщению записей совпадающих файлов журнала в измененном порядке (проблема не возникает, если файлы журнала имеют разное время последнего изменения).

- Агент Zabbix обрабатывает новые записи файла журнала один раз в *Update interval* секунд.
- Агент Zabbix не отправляет более `maxlines` строк файла журнала в секунду. Это ограничение предотвращает перегрузку сетевых и процессорных ресурсов и переопределяет значение по умолчанию, заданное параметром `MaxLinesPerSecond` в файле конфигурации агента.
- Чтобы найти требуемую строку, Zabbix будет обрабатывать в 10 раз больше новых строк, чем задано в `MaxLinesPerSecond`. Таким образом, например, если у элемента данных `log[]` или `logrt[]` *Update interval* равен 1 секунде, по умолчанию агент будет анализировать не более 200 записей файла журнала и отправлять не более 20 совпадающих записей на сервер Zabbix за одну проверку. При увеличении `MaxLinesPerSecond` в файле конфигурации агента или задании параметра `maxlines` в ключе элемента данных предел можно увеличить до 10000 проанализированных записей файла журнала и 1000 совпадающих записей, отправляемых на сервер Zabbix за одну проверку. Если *Update interval* установлен в 2 секунды, ограничения для одной проверки будут в 2 раза выше, чем при *Update interval* в 1 секунду.
- Кроме того, значения `log` и `log.count` всегда ограничены 50% размера буфера отправки агента, даже если в нем нет других значений, не относящихся к журналам. Поэтому для отправки значений `maxlines` за одно соединение (а не за несколько) параметр `BufferSize` агента должен быть не меньше `maxlines` x 2. Агент Zabbix может загружать данные во время сбора журнала и тем самым освобождать буфер, тогда как агент Zabbix 2 остановит сбор журнала до тех пор, пока данные не будут загружены и буфер не освободится; это выполняется асинхронно.
- При отсутствии элементов данных журнала весь буфер агента используется для значений, не относящихся к журналам. Когда поступают значения журнала, они при необходимости заменяют более старые значения, не относящиеся к журналам, вплоть до выделенных 50%.
- Для записей файла журнала длиной более 256kB только первые 256kB сопоставляются с регулярным выражением, а остальная часть записи игнорируется. Однако если агент Zabbix остановлен во время обработки длинной записи, внутреннее состояние агента теряется, и длинная запись может быть проанализирована снова и иначе после повторного запуска агента.
- Особое примечание для разделителей пути `\`: если `file_format` равен `file\.log`, то каталога `file` быть не должно, поскольку невозможно однозначно определить, экранирована ли `.` или это первый символ имени файла.
- Регулярные выражения для `logrt` поддерживаются только в имени файла, сопоставление регулярного выражения для каталога не поддерживается.
- На платформах UNIX элемент данных `logrt[]` становится NOTSUPPORTED, если каталог, в котором ожидается наличие файлов журнала, не существует.
- В Microsoft Windows, если каталог не существует, элемент данных не станет NOTSUPPORTED (например, если каталог указан с ошибкой в ключе элемента данных).
- Отсутствие файлов журнала для элемента данных `logrt[]` не делает его NOTSUPPORTED. Ошибки чтения файлов журнала для элемента данных `logrt[]` записываются как предупреждения в файл журнала агента Zabbix, но не делают элемент данных NOTSUPPORTED.
- Файл журнала агента Zabbix может помочь выяснить, почему элемент данных `log[]` или `logrt[]` стал NOTSUPPORTED. Zabbix может отслеживать файл журнала своего агента, кроме случаев, когда установлено `DebugLevel=4` или `DebugLevel=5`.
- Поиск вопросительного знака с помощью регулярного выражения, например `\?`, может приводить к ложным срабатываниям, если текстовый файл содержит символы NUL, поскольку Zabbix заменяет их на ?, чтобы продолжить обработку строки до символа новой строки.

Извлечение совпавшей части регулярного выражения

Иногда требуется извлечь только интересующее значение из целевого файла, а не возвращать всю строку при обнаружении совпадения с регулярным выражением.

Элементы данных журнала могут извлекать нужные значения из совпавших строк. Это достигается с помощью дополнительного параметра **output** в элементах данных `log` и `logrt`.

Параметр `output` позволяет указать «группу захвата» совпадения, которая представляет интерес.

Например:

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

должен возвращать количество записей, найденное в содержимом:

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

Будет возвращено только число, поскольку \1 ссылается на первую и единственную группу захвата: ([0-9]+).

Благодаря возможности извлекать и возвращать число это значение можно использовать для определения триггеров.

Использование параметра maxdelay

Параметр maxdelay в элементах данных типа log позволяет игнорировать некоторые более старые строки из файлов журналов, чтобы в пределах maxdelay секунд анализировались самые последние строки.

> **Warning:**
> Указание maxdelay > 0 может привести к **игнорированию важных записей файла журнала и пропущенным оповещениям**. Используйте его осторожно и только при необходимости, на свой страх и риск.

По умолчанию элементы данных для мониторинга журналов обрабатывают все новые строки, появляющиеся в файлах журналов. Однако существуют приложения, которые в некоторых ситуациях начинают записывать огромное количество сообщений в свои файлы журналов. Например, если база данных или DNS-сервер недоступны, такие приложения заполняют файлы журналов тысячами почти одинаковых сообщений об ошибках, пока нормальная работа не будет восстановлена. По умолчанию все эти сообщения добросовестно анализируются, а совпадающие строки отправляются на сервер, как настроено в элементах данных log и logrt.

Встроенная защита от перегрузки состоит из настраиваемого параметра maxlines (защищает сервер от слишком большого количества входящих совпадающих строк журнала) и ограничения 10*maxlines (защищает CPU и I/O узла сети от перегрузки агентом за одну проверку). Тем не менее, у встроенной защиты есть 2 проблемы. Во-первых, на сервер передается большое количество потенциально не слишком информативных сообщений, которые занимают место в базе данных. Во-вторых, из-за ограниченного числа строк, анализируемых в секунду, агент может отставать от самых новых записей журнала на часы. Скорее всего, вы предпочтете получать информацию о текущей ситуации в файлах журналов как можно раньше, а не разбирать старые записи в течение нескольких часов.

Решение обеих проблем — использование параметра maxdelay. Если задано maxdelay > 0, при каждой проверке измеряются количество обработанных байтов, количество оставшихся байтов и время обработки. На основе этих значений агент вычисляет оценочную задержку — сколько секунд потребуется, чтобы проанализировать все оставшиеся записи в файле журнала.

Если задержка не превышает maxdelay, агент продолжает анализ файла журнала как обычно.

Если задержка больше maxdelay, агент **игнорирует фрагмент файла журнала, "перепрыгивая" через него** к новой оценочной позиции, чтобы оставшиеся строки можно было проанализировать в пределах maxdelay секунд.

Обратите внимание, что агент даже не читает игнорируемые строки в буфер, а вычисляет приблизительную позицию, к которой нужно перейти в файле.

Факт пропуска строк файла журнала записывается в файл журнала агента примерно так:

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]"
logfile:"/home/zabbix32/test1.log" skipping 679858 bytes
(from byte 75653115 to byte 76332973) to meet maxdelay
```

Значение "to byte" является приблизительным, потому что после "прыжка" агент корректирует позицию в файле до начала строки журнала, которая может оказаться дальше в файле или раньше.

В зависимости от того, как скорость роста журнала соотносится со скоростью его анализа, вы можете не увидеть ни одного "прыжка", редкие или частые "прыжки", большие или маленькие "прыжки" или даже небольшой "прыжок" при каждой проверке. Колебания нагрузки на систему и задержки сети также влияют на расчет задержки и, следовательно, на "прыжки" вперед, чтобы успевать за параметром maxdelay.

Не рекомендуется задавать maxdelay < update interval (это может привести к частым небольшим "прыжкам").

Примечания по обработке ротации лог-файлов с copytruncate

logrt с опцией copytruncate предполагает, что разные лог-файлы содержат разные записи (по крайней мере, их временные метки различаются), поэтому MD5-суммы начальных блоков (до первых 512 байт) будут различаться. Два файла с одинаковыми MD5-суммами начальных блоков означают, что один из них является исходным, а другой — копией.

logrt с опцией copytruncate предпринимает попытки корректно обрабатывать копии лог-файлов без сообщения о дубликатах. Однако такие ситуации, как создание нескольких копий лог-файла с одинаковой временной меткой, ротация лог-файла чаще, чем интервал обновления элемента данных logrt[], а также частые перезапуски агента, не рекомендуются. Агент старается разумно обрабатывать все эти ситуации, но хорошие результаты не могут быть гарантированы при любых обстоятельствах.

Примечания о постоянных файлах для элементов данных log*[]

Назначение постоянных файлов

Когда агент Zabbix запускается, он получает список активных проверок от сервера Zabbix или прокси.
Для метрик `log*[]` он получает обработанный размер журнала и время изменения, чтобы определить, с какого места начинать мониторинг файла журнала.
В зависимости от фактического размера файла журнала и времени изменения, сообщаемых файловой системой, агент решает либо продолжить мониторинг файла журнала с обработанного размера, либо повторно проанализировать файл журнала с начала.

Работающий агент поддерживает более полный набор атрибутов для отслеживания всех контролируемых файлов журнала между проверками.
Это состояние в памяти теряется при остановке агента.

Новый необязательный параметр `persistent_dir` задает каталог для хранения этого состояния элемента данных `log[]`, `log.count[]`, `logrt[]` или `logrt.count[]` в файле.
Состояние элемента данных журнала восстанавливается из постоянного файла после перезапуска агента Zabbix.

Основной сценарий использования — мониторинг файла журнала, расположенного на зеркальной файловой системе.
До определенного момента времени файл журнала записывается на оба зеркала.
Затем зеркала разделяются.
На активной копии файл журнала по-прежнему растет, получая новые записи.
Агент Zabbix анализирует его и отправляет серверу обработанный размер журнала и время изменения.
На пассивной копии файл журнала остается без изменений, значительно отставая от активной копии.
Позже операционная система и агент Zabbix перезагружаются с пассивной копии.
Обработанный размер журнала и время изменения, которые агент Zabbix получает от сервера, могут быть недействительными для ситуации на пассивной копии.
Чтобы продолжить мониторинг файла журнала с того места, где агент остановился в момент разделения зеркала файловой системы, агент восстанавливает свое состояние из постоянного файла.

Работа агента с постоянным файлом

При запуске агент Zabbix ничего не знает о постоянных файлах. Только после получения списка активных проверок от сервера Zabbix (прокси) агент видит, что некоторые элементы данных журнала должны храниться в постоянных файлах в указанных каталогах.

Во время работы агента постоянные файлы открываются для записи (с помощью `fopen(filename, "w")`) и перезаписываются последними данными. Вероятность потери данных постоянного файла, если перезапись и сбой зеркалирования файловой системы произойдут одновременно, очень мала, поэтому специальная обработка не требуется. Запись в постоянный файл НЕ сопровождается принудительной синхронизацией с носителем данных (`fsync()` не вызывается).

Перезапись последними данными выполняется после успешной передачи серверу Zabbix соответствующей записи файла журнала или метаданных (обработанный размер журнала и время изменения). Это может происходить так часто, как при каждой проверке элемента данных, если файл журнала продолжает изменяться.

При завершении работы агента никаких специальных действий не выполняется.

После получения списка активных проверок агент помечает устаревшие постоянные файлы для удаления. Постоянный файл становится устаревшим, если:

1. Соответствующий log-элемент данных больше не отслеживается.
2. log-элемент данных перенастраивается на другое расположение `persistent_dir`, чем раньше.

Удаление выполняется с задержкой 24 часа, потому что файлы журналов в состоянии NOTSUPPORTED не включаются в список активных проверок, но позже они могут стать SUPPORTED, и их постоянные файлы будут полезны.

Если агент остановлен до истечения 24 часов, устаревшие файлы не будут удалены, так как агент Zabbix больше не получает от сервера Zabbix информацию об их расположении.

Warning:
Перенастройка `persistent_dir` log-элемента данных обратно на прежнее расположение `persistent_dir` во время остановки агента, без удаления пользователем старого постоянного файла, приведет к восстановлению состояния агента из старого постоянного файла, что вызовет пропуск сообщений или ложные оповещения.

Именование и расположение постоянных файлов

Агент Zabbix различает активные проверки по их ключам. Например, `logrt[/home/zabbix/test.log]` и `logrt[/home/zabbix/te` являются разными элементами данных. Изменение элемента данных `logrt[/home/zabbix/test.log,,,10]` в веб-интерфейсе на `logrt[/home/zabbix/test.log,,,20]` приведет к удалению элемента данных `logrt[/home/zabbix/test.log,,`

из списка активных проверок агента и созданию элемента данных `logrt[/home/zabbix/test.log,,,20]` (некоторые атрибуты переносятся при изменении в веб-интерфейсе/на сервере, но не в агенте).

Имя файла состоит из контрольной суммы MD5 ключа элемента данных с добавленной длиной ключа элемента данных, чтобы снизить вероятность коллизий. Например, состояние элемента данных `logrt[/home/zabbix50/test.log,,,,,,,,/home/z` будет храниться в постоянном файле c963ade4008054813bbc0a650bb8e09266.

Несколько элементов данных журнала могут использовать одно и то же значение `persistent_dir`.

Значение `persistent_dir` следует указывать с учетом особенностей структуры файловой системы, точек монтирования и параметров монтирования, а также конфигурации зеркалирования хранилища - постоянный файл должен находиться в той же зеркальной файловой системе, что и отслеживаемый файл журнала.

Если каталог `persistent_dir` невозможно создать или он не существует либо у агента Zabbix недостаточно прав для создания, записи, чтения или удаления файлов, элемент данных журнала получает статус `NOTSUPPORTED`.

Если во время работы агента права доступа к файлам постоянного хранилища были удалены или произошли другие ошибки (например, диск заполнен), ошибки записываются в файл журнала агента, но элемент данных журнала не переходит в состояние `NOTSUPPORTED`.

Нагрузка на I/O

Постоянный файл элемента данных обновляется после успешной отправки каждого пакета данных (содержащего данные элемента) на сервер. Например, значение `BufferSize` по умолчанию равно 100. Если элемент журнала нашел 70 совпадающих записей, то первые 50 записей будут отправлены в одном пакете, постоянный файл будет обновлен, затем оставшиеся 20 записей будут отправлены (возможно, с некоторой задержкой, когда накопится больше данных) во втором пакете, и постоянный файл будет снова обновлен.

Действия при сбое связи между агентом и сервером

Каждая совпадающая строка из элемента данных `log[]` и `logrt[]`, а также результат каждой проверки элемента данных `log.count[]` и `logrt.count[]` требует свободного слота в выделенной 50% области буфера отправки агента. Элементы буфера регулярно отправляются на сервер (или прокси), после чего слоты буфера снова становятся свободными.

Пока в выделенной области журнала в буфере отправки агента есть свободные слоты и связь между агентом и сервером (или прокси) отсутствует, результаты мониторинга журнала накапливаются в буфере отправки. Это помогает сгладить кратковременные сбои связи.

При более длительных сбоях связи все слоты журнала заполняются, и выполняются следующие действия:

- Проверки элементов данных `log[]` и `logrt[]` останавливаются. Когда связь восстанавливается и в буфере доступны свободные слоты, проверки возобновляются с предыдущей позиции. Совпадающие строки не теряются, они просто будут сообщены позже.
- Проверки `log.count[]` и `logrt.count[]` останавливаются, если `maxdelay = 0` (по умолчанию). Поведение аналогично элементам данных `log[]` и `logrt[]`, как описано выше. Обратите внимание, что это может повлиять на результаты `log.count[]` и `logrt.count[]`: например, одна проверка подсчитывает 100 совпадающих строк в файле журнала, но так как в буфере нет свободных слотов, проверка останавливается. Когда связь восстанавливается, агент подсчитывает те же 100 совпадающих строк, а также 70 новых совпадающих строк. Теперь агент отправляет count = 170, как если бы они были найдены в одной проверке.
- Проверки `log.count[]` и `logrt.count[]` с `maxdelay > 0`: если во время проверки не было "jump", поведение аналогично описанному выше. Если произошел "jump" через строки файла журнала, то позиция после "jump" сохраняется, а подсчитанный результат отбрасывается. Таким образом, агент старается не отставать от растущего файла журнала даже при сбое связи.

Обработка ошибок компиляции и выполнения регулярных выражений

Если регулярное выражение, используемое в элементе данных `log[]`, `logrt[]`, `log.count[]` или `logrt.count[]`, не может быть скомпилировано библиотекой PCRE или PCRE2, элемент данных переходит в состояние НЕПОДДЕРЖИВАЕТСЯ с сообщением об ошибке. Чтобы продолжить мониторинг элемента данных журнала, необходимо исправить регулярное выражение.

Если регулярное выражение успешно компилируется, но завершается ошибкой во время выполнения (для некоторых или всех записей журнала), элемент данных журнала остается поддерживаемым, а мониторинг продолжается. Ошибка выполнения записывается в файл журнала агента Zabbix (без записи из файла журнала).

Частота записи ограничена одной ошибкой выполнения за одну проверку, чтобы агент Zabbix мог отслеживать собственный файл журнала. Например, если анализируются 10 записей и для 3 записей возникает ошибка выполнения регулярного выражения, в журнале агента появляется одна запись.

Исключение: если `MaxLinesPerSecond=1` и `update interval=1` (что позволяет анализировать только одну запись за проверку), ошибки выполнения регулярного выражения не записываются.

zabbix_agentd записывает ключ элемента данных в случае ошибки выполнения, а zabbix_agent2 записывает идентификатор элемента данных, чтобы помочь определить, в каком элементе данных журнала возникают ошибки выполнения. В случае ошибок выполнения рекомендуется переработать регулярное выражение.

Простая проверка

Обзор

Простые проверки обычно используются для удаленных проверок служб без агента.

Обратите внимание, что для простых проверок агент Zabbix не нужен. За обработку простых проверок отвечает сервер Zabbix/прокси (выполнение внешних подключений и т. д.).

Примеры использования простых проверок:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
Поля *User name* и *Password* (ограничены 255 символами) в конфигурации элемента данных простой проверки используются для элементов мониторинга VMware; в остальных случаях игнорируются.

Поддерживаемые проверки

Ключи элементов данных перечислены без необязательных параметров и дополнительной информации. Щелкните по ключу элемента данных, чтобы увидеть полные сведения.

См. также ключи элементов данных для мониторинга VMware.

Ключ элемента данных	Описание
icmpping	Доступность узла сети по ICMP ping.
icmppingloss	Процент потерянных пакетов.
icmppingretry	Доступность узла сети по ICMP ping с повторными попытками.
icmppingsec	Время отклика ICMP ping.
net.tcp.service	Проверяет, запущена ли служба и принимает ли она TCP-соединения.
net.tcp.service.perf	Проверяет производительность службы TCP.
net.udp.service	Проверяет, запущена ли служба и отвечает ли она на запросы UDP.
net.udp.service.perf	Проверяет производительность службы UDP.

Подробности ключа элемента данных

Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

icmpping[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>,<опции>]

 Доступность хоста через пинг по ICMP.
 Возвращаемое значение: *0* — ошибка при пинге по ICMP; *1* — успешный пинг по ICMP.

Параметры:

- **цель** — IP или DNS имя хоста;
- **пакеты** — количество пакетов;
- **интервал** — время между последовательными пакетами в миллисекундах;
- **размер** — размер пакета в байтах;
- **время ожидания** — время ожидания в миллисекундах;
- **опции** — используется для разрешения перенаправления: если пусто (значение по умолчанию), перенаправленные ответы рассматриваются как целевой хост выключен; если установлено значение *allow_redirect*, перенаправленные ответы рассматриваются как целевой хост включён.

Смотрите также таблицу значений по умолчанию.

Комментарий:

- Если возвращается хотя бы один пакет, заданный параметром , элемент данных возвращает 1. Если не возвращается ни один из пакетов, элемент данных возвращает 0.

Пример:

```
icmpping[,4] #                          ,              1.
```

icmppingloss[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>,<опции>]

 Процентное отношение потерянных пакетов.
 Возвращаемое значение: *Число с плавающей точкой*.

Параметры:

- **цель** — IP или DNS имя хоста;
- **пакеты** — количество пакетов;
- **интервал** — время между последовательными пакетами в миллисекундах;
- **размер** — размер пакета в байтах;
- **время ожидания** — время ожидания в миллисекундах;
- **опции** — используется для разрешения перенаправления: если пусто (значение по умолчанию), перенаправленные ответы рассматриваются как целевой хост выключен; если установлено значение *allow_redirect*, перенаправленные ответы рассматриваются как целевой хост включён.

Смотрите также таблицу значений по умолчанию.

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

 Доступность узла сети по ICMP ping с повторными попытками. Если первый пакет успешно доставлен, проверка останавливается; если пакет не проходит, попытки повторяются до достижения максимального числа попыток, заданного в параметре `retries`. Этот элемент данных полезен для уменьшения количества пакетов, передаваемых по сети.
 Возвращаемое значение: *0* — ICMP ping неуспешен; *1* — ICMP ping успешен.

Параметры:

- **target** — IP-адрес или DNS-имя узла сети;
- **retries** — количество повторных попыток ping целевого узла, не включая первую попытку (0 или больше; по умолчанию 1);
- **backoff** — множитель, на который увеличивается время ожидания при каждом последующем запросе (диапазон 1.0-5.0; по умолчанию 1.0);
- **size** — размер пакета в байтах;
- **timeout** — тайм-аут в миллисекундах;
- **options** — используется для разрешения перенаправления: если пусто (значение по умолчанию), ответы с перенаправлением обрабатываются так, как будто целевой узел сети недоступен; если установлено значение *allow_redirect*, ответы с перенаправлением обрабатываются так, как будто целевой узел сети доступен.

См. также таблицу значений по умолчанию.

icmppingsec[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>,<режим>,<опции>]

 Время ответа на пинг по ICMP (в секундах).
 Возвращаемое значение: *Число с плавающей точкой*.

Параметры:

- **цель** — IP или DNS имя хоста;
- **пакеты** — количество пакетов;
- **интервал** — время между последовательными пакетами в миллисекундах;
- **размер** — размер пакета в байтах;
- **время ожидания** — время ожидания в миллисекундах;
- **режим** — один из min, max или avg (по умолчанию);
- **опции** — используется для разрешения перенаправления: если пусто (значение по умолчанию), перенаправленные ответы рассматриваются как целевой хост выключен; если установлено значение *allow_redirect*, перенаправленные ответы рассматриваются как целевой хост включён.

Комментарии:

- Потерянные пакеты либо пакеты с превышенным временем ожидания при вычислениях не учитываются;
- Если хост недоступен (превышено время ожидания), элемент данных вернёт 0;
- Если возвращаемое значение меньше 0.0001 секунд, то оно будет выставлено в 0.0001 секунд;
- Смотрите также таблицу значений по умолчанию.

net.tcp.service[сервис,<ip>,<порт>]

 Проверка того, что сервис запущен и принимает TCP подключения.
 Возвращаемое значение: *0* — сервис недоступен; *1* — сервис работает.

Параметры:

- **сервис** — возможные значения: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (смотрите подробности);
- **ip** — IP или DNS имя хоста (по умолчанию используется IP/DNS узла сети);
- **порт** — номер порта (по умолчанию используется стандартный для сервиса номер порта).

Комментарии:

- Обратите внимание, для сервиса *tcp* обязательно нужно указывать порт;
- Эти проверки могут привести к дополнительным записям в файлах журналов системных демонов (обычно журналируются сессии SMTP и SSH);
- Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как обходной вариант решения, пожалуйста, используйте `net.tcp.service[tcp,<ip>, ]` для подобных проверок.

Пример:

```
net.tcp.service[ftp,,45] #                                    FTP        45     TCP
```

Attention:
Если SELinux работает в принудительном режиме (enforced mode), пользовательские простые проверки TCP/UDP могут быть заблокированы политикой. Чтобы проверить и разрешить новое исходящее соединение, просмотрите отказы аудита: `grep denied /var/log/audit/audit.log`

net.tcp.service.perf[сервис,<ip>,<порт>]

 Проверка производительности TCP сервиса.
 Возвращаемое значение: *Число с плавающей точкой*: *0.000000* — сервис недоступен; *секунды* — количество секунд, потребовавшихся для подключения к сервису.

Параметры:

- **сервис** — возможные значения: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (смотрите подробности);
- **ip** — IP или DNS имя (по умолчанию используется IP/DNS узла сети);
- **порт** — номер порта (по умолчанию используется стандартный для сервиса номер порта).

Комментарии:

- Обратите внимание, для сервиса *tcp* обязательно нужно указывать порт;
- Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как обходной вариант решения, пожалуйста, используйте `net.tcp.service[tcp,<ip>, ]` для подобных проверок.

Пример:

```
net.tcp.service.perf[ssh] #                                          SSH      .
```

net.udp.service[сервис,<ip>,<порт>]

 Проверка, запущен ли сервис и отвечает ли на UDP запросы.
 Возвращаемое значение: *0* — сервис недоступен; *1* — сервис работает.

Параметры:

- **сервис** — возможные значения: *ntp* (смотрите подробности);
- **ip** — IP адрес или DNS имя (по умолчанию используется IP/DNS узла сети);
- **порт** — номер порта (по умолчанию используется стандартный для сервиса номер порта).

Пример:

```
net.udp.service[ntp,,45] #                                     NTP        45     UDP.
```

net.udp.service.perf[сервис,<ip>,<порт>]

 Проверка производительности UDP сервиса.
 Возвращаемое значение: *Число с плавающей точкой*: *0.000000* — сервис недоступен; *секунды* — количество секунд, потребовавшихся на ожидание ответа от сервиса.

Параметры:

- **сервис** — возможные значения: *ntp* (смотрите подробности);
- **ip** — IP адрес или DNS имя (по умолчанию используется IP/DNS узла сети);
- **порт** — номер порта (по умолчанию используется стандартный для сервиса номер порта).

Пример:

```
net.udp.service.perf[ntp] #                                          NTP      .
```

Attention:
Для поддержки SourceIP в простых проверках LDAP (например, `net.tcp.service[ldap]`) требуется OpenLDAP версии 2.6.1 или выше.

Обработка тайм-аута

Гибкие тайм-ауты элементов данных, хотя и поддерживаются для простых проверок, не применяются к `icmpping*` и элементам данных VMware. См. Поддержка гибкого тайм-аута.

ICMP-пинги

Zabbix использует внешнюю утилиту **fping** для обработки ICMP-пингов (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**).

Установка

fping не входит в состав Zabbix и его необходимо установить отдельно:

- На различных Unix-платформах пакет fping доступен в стандартных репозиториях, но не установлен по умолчанию. В этом случае вы можете использовать менеджер пакетов для установки fping.
- Zabbix предоставляет пакеты fping для RHEL и его производных. Обратите внимание, что эти пакеты предоставляются без официальной поддержки.
- fping также можно собрать из исходного кода.

Конфигурация

Укажите расположение fping в параметре *FpingLocation* файла конфигурации Zabbix server/proxy (или в параметре *Fping6Location* при использовании адресов IPv6).

fping должен быть доступен для выполнения пользователем, от имени которого запущен Zabbix server/proxy, и у этого пользователя должны быть достаточные права.

См. также: Известные проблемы при обработке простых проверок с версиями fping ниже 3.10.

Значения по умолчанию

Значения по умолчанию, ограничения и описание параметров проверки ICMP:

Параметр	Единица измерения	Описание	Флаг fping	Значения по умолчанию задаются		Допустимые ограничения в Zabbix	
				fping	**Zabbix**	**мин.**	**макс.**
packets	количество	Количество пакетов запросов, отправляемых целевому узлу сети	-C		3	1	10000
interval	миллисекунды	Время ожидания между последовательными пакетами, отправляемыми отдельному целевому узлу сети	-p	1000		20	не ограничено
size	байты	Размер пакета в байтах 56 байт на x86, 68 байт на x86_64	-b	56 или 68		24	65507
timeout	миллисекунды	**fping v3.x** - время ожидания после отправки последнего пакета, на которое влияет флаг −C **fping v4.x** - индивидуальное время ожидания для каждого пакета	-t	**fping v3.x** - 500 **fping v4.x** и более поздние версии - наследуется от флага −p, но не превышает 2000		50	не ограничено

Параметр	Единица измерения	Описание	Флаг fping	Значения по умолчанию задаются		Допустимые ограничения в Zabbix	
retries	количество	Количество повторных попыток проверки доступности целевого узла сети, не включая первую попытку	-r	3	1	0	не ограничено
backoff factor	количество	Число, на которое умножается время ожидания при каждом последующем запросе	-B	1.5	1.0	1.0	5.0

Значения по умолчанию могут незначительно отличаться в зависимости от платформы и версии.

Кроме того, Zabbix использует параметры fping `-i interval ms` (не путайте их с параметром элемента данных `interval`, указанным в таблице выше, которому соответствует параметр fping `-p`) и `-S source IP address` (или `-I` в старых версиях fping). Эти параметры определяются автоматически путем выполнения проверок с различными комбинациями параметров. Zabbix пытается определить минимальное значение в миллисекундах, которое fping допускает для параметра `-i`, проверяя три значения: 0, 1 и 10. Первое успешно обработанное значение затем используется для последующих проверок ICMP. Этот процесс выполняется каждым процессом ICMP pinger отдельно.

Автоматически определенные параметры fping становятся недействительными каждый час и определяются заново при следующей попытке выполнить проверку ICMP. Установите значение DebugLevel>=4, чтобы просматривать сведения об этом процессе в файле журнала сервера или прокси.

Zabbix записывает IP-адреса, которые необходимо проверить с помощью любых ключей `icmpping*`, во временный файл, который затем передается fping. Если элементы данных имеют разные параметры ключа, в один файл записываются только элементы с идентичными параметрами ключа. Все IP-адреса, записанные в один файл, проверяются fping параллельно, поэтому процесс ICMP pinger Zabbix тратит фиксированное количество времени независимо от числа IP-адресов в файле.

Ключи элементов данных для мониторинга VMware

Список ключей элементов данных для мониторинга VMware перенесён в раздел Мониторинг VMware.

SNMP агент

Обзор

Возможно, вы захотите использовать SNMP-мониторинг на таких устройствах, как принтеры, сетевые коммутаторы, маршрутизаторы или ИБП, которые обычно поддерживают SNMP, и на которых было бы непрактично пытаться устанавливать полноценные операционные системы и агенты Zabbix.

Чтобы иметь возможность получать данные, предоставляемые SNMP-агентами на этих устройствах, сервер Zabbix должен быть изначально настроен с поддержкой SNMP путем указания флага `--with-net-snmp`. Также рекомендуется установить файлы MIB, чтобы значения элементов данных отображались в правильном формате. Без файлов MIB могут возникать проблемы с форматированием, например отображение значений в HEX вместо UTF-8 или наоборот.

Проверки SNMP выполняются только по протоколу UDP.

Сервер Zabbix и демоны прокси записывают в журнал строки, подобные следующей, если получают некорректный SNMP-ответ:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Хотя они не охватывают все проблемные случаи, они полезны для выявления отдельных SNMP-устройств, для которых следует отключить объединенные запросы.

Сервер/прокси Zabbix будет повторять попытку до 5 раз для элементов данных SNMP `walk` и `get`. Механизм повторных попыток не применяется к сбоям разрешения DNS.

Для устаревших SNMP-проверок (один OID-номер или строка) сервер/прокси Zabbix выполнит как минимум одну повторную попытку после неудачного запроса: либо через механизм повторов библиотеки SNMP, либо через внутренний механизм объединенной обработки.

Warning:
Если вы мониторите устройства SNMPv3, убедитесь, что msgAuthoritativeEngineID (также известный как snmpEngineID или "Engine ID") никогда не используется двумя устройствами одновременно. Согласно RFC 2571 (раздел 3.1.1.1) он должен быть уникальным для каждого устройства.

Warning:
RFC3414 требует, чтобы устройства SNMPv3 сохраняли свои engineBoots. Некоторые устройства этого не делают, из-за чего их SNMP-сообщения после перезапуска отбрасываются как устаревшие. В такой ситуации кэш SNMP необходимо вручную очистить на сервере/прокси (с помощью -R snmp_cache_reload) либо перезапустить сервер/прокси.

Note:
Zabbix кэширует сопоставления SNMPv3 EngineID → IP и повторно использует кэшированные EngineID для последующих проверок вместо отправки probe каждый раз, уменьшая сетевой трафик. Если EngineID нельзя повторно использовать, будет выполнена повторная попытка с probe для обнаружения нового EngineID.

Настройка мониторинга по SNMP

Для начала мониторинга устройства по SNMP нужно выполнить следующие шаги:

Шаг 1

Определите строку SNMP (или OID) элемента данных, который вы хотите отслеживать.

Чтобы получить список строк SNMP, используйте команду `snmpwalk` (часть программного обеспечения net-snmp, которое должно быть установлено вместе с Zabbix) или аналогичный инструмент:

```
snmpwalk -v 2c -c public <host IP> .
```

Поскольку 2c здесь обозначает версию SNMP, вы также можете заменить его на 1, чтобы указать на устройстве SNMP Version 1.

В результате вы получите список строк SNMP и их последнее значение. Если этого не происходит, возможно, SNMP 'community' отличается от стандартного 'public', и тогда вам нужно будет выяснить, какое именно значение используется.

Затем вы можете просмотреть список, пока не найдете строку, которую хотите отслеживать, например, если вы хотите отслеживать входящие байты на вашем коммутаторе на порту 3, вы будете использовать строку `IF-MIB::ifHCInOctets.3` из этой строки:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Теперь вы можете использовать команду `snmpget`, чтобы определить числовой OID для `IF-MIB::ifHCInOctets.3`:

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Обратите внимание, что последнее число в строке — это номер порта, который вы хотите отслеживать. См. также: Dynamic indexes.

В результате вы получите что-то вроде следующего:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

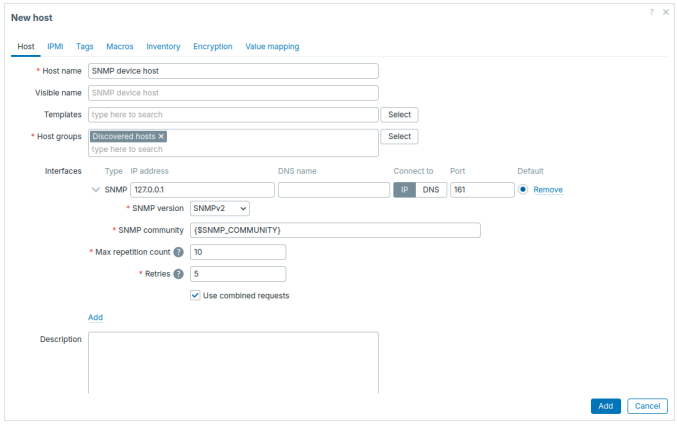
И снова, последнее число в OID — это номер порта.

Note:
Некоторые из наиболее часто используемых SNMP OID автоматически преобразуются в числовое представление Zabbix.

В последнем примере выше тип значения — "Counter64", который внутренне соответствует типу ASN_COUNTER64. Полный список поддерживаемых типов: ASN_COUNTER, ASN_COUNTER64, ASN_UINTEGER, ASN_UNSIGNED64, ASN_INTEGER, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS, ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR и ASN_OBJECT_ID. Эти типы примерно соответствуют "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" в выводе **snmpget**, но также могут отображаться как "STRING", "Hex-STRING", "OID" и другие, в зависимости от наличия подсказки отображения.

Шаг 2

Создайте узел сети, соответствующий устройству.



Добавьте SNMP-интерфейс для узла сети:

1. Введите IP-адрес/DNS-имя и номер порта.
2. Выберите *версию SNMP* из раскрывающегося списка.
3. Добавьте учетные данные интерфейса в зависимости от выбранной версии SNMP:
    - SNMPv1, v2 требуют только community (обычно 'public').
    - SNMPv3 требует более специфичных параметров; см. SNMPv3.
4. Укажите максимальное значение повторений (по умолчанию: 10) для нативных SNMP bulk-запросов (GetBulkRequest-PDU); только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. Обратите внимание, что слишком большое значение может привести к тайм-ауту проверки агента SNMP.
5. Отметьте флажок *Использовать объединенные запросы*, чтобы разрешить объединенную обработку SNMP-запросов (не относится к нативным SNMP bulk-запросам "walk" и "get").

Вы можете использовать один из предоставленных шаблонов SNMP, который автоматически добавит набор элементов данных. Перед использованием шаблона убедитесь, что он совместим с узлом сети.

Нажмите *Добавить*, чтобы сохранить узел сети.

SNMPv3

Для SNMPv3 требуются следующие параметры:

- *Context name* - Укажите имя контекста для идентификации элемента данных в подсети SNMP.
В этом поле подставляются макросы пользователя.
- *Security name* - Укажите имя безопасности.
В этом поле подставляются макросы пользователя.
- *Security level* - Выберите уровень безопасности:
    - *noAuthNoPriv* - не используются ни протоколы аутентификации, ни протоколы конфиденциальности;
    - *AuthNoPriv* - используется протокол аутентификации, протокол конфиденциальности не используется;
    - *AuthPriv* - используются и протокол аутентификации, и протокол конфиденциальности.
- *Authentication protocol* - Выберите протокол аутентификации: *MD5*, *SHA1*; для net-snmp 5.8 и новее - *SHA224*, *SHA256*, *SHA384* или *SHA512*.
- *Authentication passphrase* - Укажите парольную фразу аутентификации.
В этом поле подставляются макросы пользователя.
- *Privacy protocol* - Выберите протокол конфиденциальности: *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) или *AES256C* (Cisco).
См. примечания о поддержке протокола конфиденциальности.
- *Privacy passphrase* - Укажите парольную фразу конфиденциальности.
В этом поле подставляются макросы пользователя.

В случае неверных учетных данных SNMPv3 (security name, authentication protocol/passphrase, privacy protocol):

- Zabbix получает ERROR от net-snmp, за исключением случая неверной *Privacy passphrase*, когда Zabbix получает ошибку TIMEOUT от net-snmp.
- Доступность интерфейса SNMP переключится на красный цвет (недоступен).

Note:
Изменения в *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* или *Privacy passphrase*, внесенные без изменения *Security name*, обычно применяются автоматически при обновлении соответствующего интерфейса SNMPv3 в Zabbix. В случаях, когда также изменяется *Security name*, все параметры будут обновлены немедленно.

Поддержка протоколов конфиденциальности

В зависимости от вашей операционной системы и конфигурации net-snmp некоторые протоколы конфиденциальности могут быть недоступны:

- В некоторых более новых операционных системах (например, RHEL9) поддержка DES для пакета net-snmp отключена.
- Протоколы шифрования AES192 и более сильные не поддерживаются "из коробки" в операционных системах старше, чем RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5.

Чтобы проверить, поддерживает ли библиотека net-snmp AES192+, используйте один из следующих вариантов:

1. net-snmp-config:

```
net-snmp-config --configure-options
```

Если вывод содержит --enable-blumenthal-aes, поддерживается AES192+.

Обратите внимание, что net-snmp-config входит в состав пакета разработки для SNMP (libsnmp-dev для Debian/Ubuntu, net-snmp-devel для CentOS/RHEL/OL/SUSE) и может не быть установлен по умолчанию.

2. snmpget:

```
snmpget -v 3 -x AES-256
```

Если вывод содержит Invalid privacy protocol specified after -3x flag: AES-256, AES192+ не поддерживается.
Если вывод содержит No hostname specified., AES192+ не поддерживается.

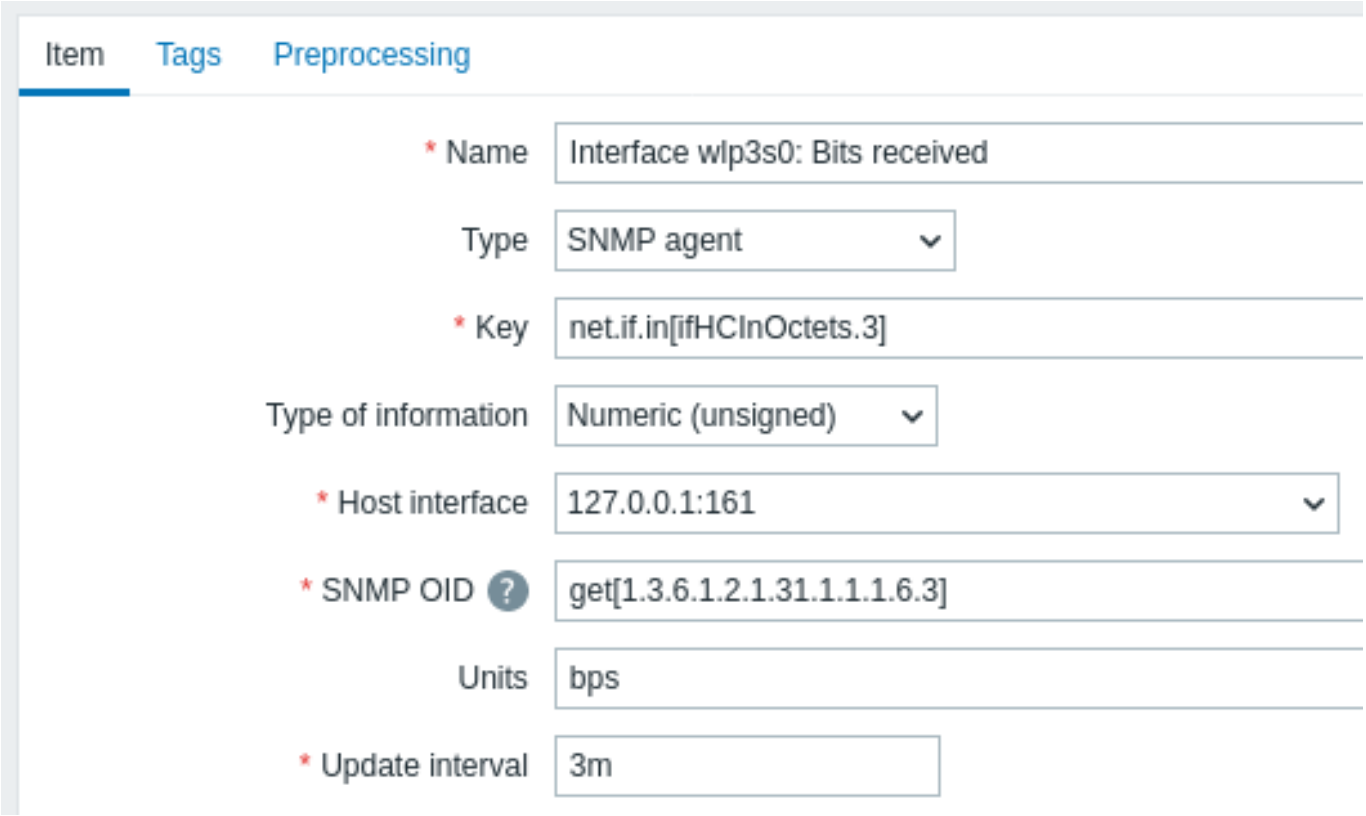
Если ваша библиотека net-snmp не поддерживает протоколы AES192 и выше, перекомпилируйте net-snmp с опцией --enable-blumenthal-aes, затем перекомпилируйте сервер Zabbix, указав опцию --with-net-snmp=/home/user/yourcustomne

Шаг 3

Создайте элемент данных для мониторинга.

Теперь вернитесь в Zabbix и нажмите *Элементы данных* для узла сети SNMP, который вы создали ранее. В зависимости от того, использовали ли вы шаблон при создании узла сети, у вас будет либо список элементов данных SNMP, связанных с узлом сети, либо пустой список. Мы предполагаем, что вы создадите элемент данных самостоятельно, используя информацию, которую только что получили с помощью snmpwalk и snmpget, поэтому нажмите *Создать элемент данных*.

Заполните обязательные параметры в форме нового элемента данных:



Параметр	Описание
Имя	Введите имя элемента данных.
Тип	Выберите здесь **SNMP agent**.
Ключ	Введите содержательный ключ.

Параметр	Описание
Интерфейс узла сети	Обязательно выберите интерфейс SNMP, например интерфейс коммутатора или маршрутизатора.

Параметр	Описание
SNMP OID	Используйте один из поддерживаемых форматов для ввода значения или значений OID: **walk[OID1,OID2,...]** - получение поддерева значений. Например: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Этот параметр использует встроенные массовые запросы SNMP (GetBulkRequest-PDU) в асинхронном режиме. Параметры времени ожидания для этого элемента данных можно задать в форме настройки элемента данных. Рекомендуется установить небольшое значение времени ожидания, чтобы избежать длительных задержек, если устройство недоступно, поскольку при истечении времени ожидания или сбое предыдущей попытки будет выполнено до 5 повторных попыток (например, время ожидания 3 секунды может привести к ожиданию в течение 15 секунд). Этот элемент данных можно использовать как ведущий, а зависимые элементы данных - для извлечения данных из ведущего с помощью предварительной обработки. В одном обходе SNMP можно указать несколько OID, например `walk[OID1,OID2,...]`, чтобы асинхронно обрабатывать по одному OID за раз. Если массовый запрос не возвращает результатов, предпринимается попытка получить одну запись без массового запроса. В качестве параметров поддерживаются имена MIB, поэтому `walk[1.3.6.1.2.1.2.2.1.2]` и `walk[ifDescr]` вернут одинаковый результат. Если указано несколько OID/MIB, например `walk[ifDescr,ifType,ifPhysAddress]`, результатом будет объединенный список. Запросы GetBulk используются с интерфейсами SNMPv2 и v3, а для интерфейсов SNMPv1 используется GetNext; максимальное количество повторений для массовых запросов настраивается на уровне интерфейса. Параметр максимального количества повторений влияет на массовые запросы, определяя максимальное число OID, возвращаемых в одном массовом ответе. Большее значение приводит к увеличению размера массовых ответов и уменьшает количество необходимых передач. Однако не все устройства поддерживают очень большие значения, что может привести к проблемам. Этот элемент данных возвращает результат работы утилиты snmpwalk с параметрами -Oe -Ot -On. Этот элемент данных можно использовать как ведущий элемент данных в обнаружении SNMP. **get[OID]** - асинхронное получение *одного* значения. Например: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Параметры времени ожидания для этого элемента данных можно задать в форме настройки элемента данных. Рекомендуется установить небольшое значение времени ожидания, чтобы избежать длительных задержек, если устройство недоступно, поскольку при истечении времени ожидания или сбое предыдущей попытки будет выполнено до 5 повторных попыток (например, время ожидания 3 секунды может привести к ожиданию в течение 15 секунд). **OID** - (устаревший вариант) введите один текстовый или числовой OID для синхронного получения одного значения, при необходимости в сочетании с другими значениями. Например: `1.3.6.1.2.1.31.1.1.1.6.3`. Для этого варианта время ожидания проверки элемента данных будет равно значению, заданному в файле конфигурации сервера. Для повышения производительности **рекомендуется** использовать элементы данных `walk[OID]` и `get[OID]`. Все элементы данных `walk[OID]` и `get[OID]` выполняются асинхронно - получать ответ на один запрос до начала других проверок не требуется. Разрешение DNS также выполняется асинхронно. Максимальное количество одновременно выполняемых асинхронных проверок равно 1000 (определяется параметром MaxConcurrentChecksPerPoller). Количество асинхронных опросчиков SNMP определяется параметром StartSNMPPollers. Обратите внимание, что для статистики сетевого трафика, возвращаемой любым из методов, на вкладке *Предварительная обработка* необходимо добавить шаг *Изменение в секунду*; в противном случае вместо последнего изменения вы получите накопленное значение с устройства SNMP.

Параметр	Описание

Все обязательные поля ввода отмечены красной звездочкой.

Теперь сохраните элемент данных и перейдите в раздел *Мониторинг* > *Последние данные*, чтобы просмотреть данные SNMP.

Пример 1

Общий пример:

Параметр	Описание
OID	1.2.3.45.6.7.8.0 (или .1.2.3.45.6.7.8.0)
Ключ	<Уникальная строка, которая используется как ссылка в триггерах> Например, «my_param».

Обратите внимание, что OID можно задать в числовом или строковом представлении. Тем не менее, в некоторых случаях строковый OID должен быть сконвертирован в числовое представление. Для этого можно использовать утилиту snmpget:

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

Пример 2

Мониторинг времени работы:

Параметр	Описание
OID	MIB::sysUpTime.0
Ключ	router.uptime
Тип информации	Числовой (с плавающей точкой)
Единица измерения	uptime
Множитель	0.01

Собственные массовые запросы SNMP

Элемент данных **walk[OID1,OID2,...]** позволяет использовать собственную функциональность SNMP для массовых запросов (GetBulkRequest-PDU), доступную в SNMP версий 2/3.

Запрос GetBulk в SNMP выполняет несколько запросов GetNext и возвращает результат в одном ответе. Это можно использовать как для обычных элементов данных SNMP, так и для обнаружения SNMP, чтобы минимизировать количество сетевых обменов.

Элемент данных SNMP **walk[OID1,OID2,...]** можно использовать как главный элемент данных, который собирает данные одним запросом, с зависимыми элементами данных, разбирающими ответ по мере необходимости с помощью предобработки.

Обратите внимание, что использование собственных массовых запросов SNMP не связано с возможностью объединения запросов SNMP. Это собственный способ Zabbix объединять несколько запросов SNMP (см. следующий раздел).

Для массовых элементов данных SNMP выполняется до пяти повторных попыток, чтобы избежать сбоя при потере одного из пакетов. Время ожидания для элементов данных SNMP с `get` и `walk` (задаваемое в форме настройки элемента данных) устанавливается для всего сеанса. Время ожидания применяется независимо от того, получены ли данные полностью; если данные получены частично (например, данные успешно собраны только для одного из нескольких OID), элемент данных становится неподдерживаемым с сообщением «Получены только частичные данные». Если время ожидания истекло, выполняется повторная попытка, время ожидания сбрасывается, а последний запрос отправляется повторно, что позволяет сеансу продолжить работу с последнего запроса, если один пакет потерян или поступил слишком поздно. Рекомендуется установить небольшое значение времени ожидания, чтобы избежать длительных задержек, если устройство недоступно, поскольку при истечении времени ожидания или сбое предыдущих попыток будет выполнено до 5 повторных попыток (например, время ожидания в 3 секунды может привести к ожиданию в течение 15 секунд).

Внутреннее устройство комбинированной обработки

Сервер Zabbix и прокси могут запрашивать у устройств SNMP несколько значений в одном запросе. Это влияет на несколько типов элементов данных SNMP:

- обычные элементы данных SNMP
- элементы данных SNMP с динамическими индексами

- правила обнаружения низкоуровневого обнаружения SNMP

Все элементы данных SNMP на одном интерфейсе с идентичными параметрами планируются для одновременного опроса. Первые два типа элементов данных опрашиваются поллерами пакетами не более чем по 128 элементов данных, тогда как правила обнаружения низкого уровня обрабатываются по отдельности, как и раньше.

На более низком уровне для получения значений выполняются два вида операций: получение нескольких указанных объектов и обход дерева OID.

Для «получения» используется GetRequest-PDU, содержащий не более 128 привязок переменных. Для «обхода» в SNMPv1 используется GetNextRequest-PDU, а в SNMPv2 и SNMPv3 — GetBulkRequest с полем «max-repetitions», равным не более 128.

Таким образом, преимущества комбинированной обработки для каждого типа элементов данных SNMP выглядят следующим образом:

- обычные элементы данных SNMP получают преимущества от улучшений «получения»;
- элементы данных SNMP с динамическими индексами получают преимущества как от улучшений «получения», так и от улучшений «обхода»: «получение» используется для проверки индексов, а «обход» — для построения кэша;
- правила обнаружения низкого уровня SNMP получают преимущества от улучшений «обхода».

Однако существует техническая проблема: не все устройства способны возвращать 128 значений в одном запросе. Некоторые устройства всегда возвращают корректный ответ, но другие либо отвечают ошибкой «tooBig(1)», либо вообще не отвечают, если потенциальный ответ превышает определенный предел.

Чтобы определить оптимальное количество объектов для запроса к конкретному устройству, Zabbix использует следующую стратегию. Сначала он осторожно запрашивает одно значение в одном запросе. Если запрос выполнен успешно, он запрашивает два значения в одном запросе. Если и этот запрос выполнен успешно, он запрашивает три значения и продолжает аналогичным образом, умножая количество запрашиваемых объектов на 1,5. В результате получается следующая последовательность размеров запросов: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Однако если устройство отказывается возвращать корректный ответ, например для 42 переменных, Zabbix выполняет два действия.

Во-первых, для текущего пакета элементов данных он уменьшает вдвое количество объектов в одном запросе и запрашивает 21 переменную. Если устройство доступно, запрос в подавляющем большинстве случаев должен выполниться успешно, поскольку ранее было установлено, что 28 переменных обрабатываются корректно, а 21 — значительно меньше этого значения. Однако если и этот запрос завершается ошибкой, Zabbix переходит к получению значений по одному. Если и на этом этапе запрос завершается ошибкой, устройство определенно не отвечает, и размер запроса здесь ни при чем.

Во-вторых, для последующих пакетов элементов данных Zabbix начинает с последнего успешно обработанного количества переменных — в нашем примере с 28 — и продолжает увеличивать размер запроса на 1 до достижения предела. Например, если предположить, что максимальный размер ответа составляет 32 переменные, последующие запросы будут иметь размеры 29, 30, 31, 32 и 33. Последний запрос завершится ошибкой, и Zabbix больше никогда не будет отправлять запрос размером 33. С этого момента Zabbix будет запрашивать для этого устройства не более 32 переменных.

Если большие запросы с таким количеством переменных завершаются ошибкой, это может означать одно из двух. Точные критерии, по которым устройство ограничивает размер ответа, неизвестны, но мы пытаемся приблизительно определить их по количеству переменных. Первая возможность заключается в том, что это количество переменных находится примерно на уровне фактического предела размера ответа устройства в общем случае: иногда ответ меньше предела, а иногда больше него. Вторая возможность заключается в том, что UDP-пакет в одном из направлений был просто потерян. Поэтому, если Zabbix получает ошибку при выполнении запроса, он уменьшает максимальное количество переменных, чтобы перейти в более комфортный для устройства диапазон, но делает это не более двух раз.

В приведенном выше примере, если запрос с 32 переменными случайно завершится ошибкой, Zabbix уменьшит количество до 31. Если и этот запрос завершится ошибкой, Zabbix уменьшит количество до 30. Однако Zabbix не будет уменьшать количество ниже 30, поскольку предположит, что дальнейшие ошибки вызваны потерей UDP-пакетов, а не ограничением устройства.

Если же устройство не может корректно обрабатывать комбинированные запросы по другим причинам и описанная выше эвристика не работает, для каждого интерфейса предусмотрен параметр «Использовать комбинированные запросы», позволяющий отключить комбинированные запросы для этого устройства.

Если комбинированные запросы приводят к частичным или некорректным ответам, из-за которых неправильно рассчитываются значения в секунду (дельта), например появляются мнимые скачки счетчиков интерфейсов, отключите параметр *Использовать комбинированные запросы* для соответствующего интерфейса, чтобы принудительно выполнять отдельные запросы для каждого элемента данных; это часто предотвращает ложные скачки. Также можно использовать асинхронные элементы данных `get[]` или `walk[]`, которые выполняются асинхронно и не подпадают под пакетную обработку параметра *Использовать комбинированные запросы* на уровне интерфейса. Их можно применять вместо

устаревших синхронных проверок OID, чтобы избежать проблем, связанных с комбинированными запросами. Для выявления затронутых устройств обратитесь к записям в журнале сервера/прокси, аналогичным показанной в разделе Обзор.

Кроме того, если интерфейс часто становится недоступным, может потребоваться увеличить параметр `UnavailableDelay` в файлах конфигурации сервера Zabbix или прокси Zabbix, чтобы уменьшить частоту запросов. Элементы данных могут стать неподдерживаемыми, если во время обнаружения или обхода OID получены неполные данные.

Динамические индексы

Обзор

Хотя вы можете найти требуемый номер индекса (например, сетевого интерфейса) среди SNMP OID'ов, иногда вы не можете полностью положиться на то, что номер индекса всегда останется таким же.

Индексы могут быть динамическими — они могут изменяться время от времени и, как следствие, ваш элемент данных может перестать работать.

Чтобы избежать этого, имеется возможность указать OID, который принимает во внимание возможность изменения номера индекса.

Например, если вам необходимо получить значение индекса, который относится к **ifInOctets**, которое соответствует интерфейсу **GigabitEthernet0/1** на устройстве Cisco, используйте следующий OID:

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

Синтаксис

Для OID используется специальный синтаксис:

<OID данных>["index","<базовый OID индекса>","<искомая строка>"]

Параметр	Описание
OID данных	Основной OID, который используется для получения данных по элементу данных.
index	Метод обработки. В настоящее время поддерживается только один метод: **index** — поиск индекса и присоединение его к OID'у данных
базовый OID индекса	Этот OID будет просматриваться для получения значения индекса, соответствующего заданной строке.
искомая строка	Строка, используемая для точного совпадения со значением при выполнении поиска. Регистрозависимо.

Пример

Получение использования памяти процессом *apache*.

Если используется этот синтаксис OID:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

номер индекса будет искаться здесь:

```
...
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
...
```

Теперь у нас есть индекс, равный 5388. Индекс будет присоединён к OID'у данных, чтобы получить интересующее нас значение:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Кэширование поиска индексов

При запросе динамического индекса элемента данных Zabbix извлекает и кэширует всю таблицу SNMP под базовым OID, даже если совпадение будет найдено ранее. Это делается на случай, если другой элемент данных в дальнейшем будет ссылаться на тот же самый базовый OID — Zabbix будет искать индекс в кэше вместо ещё одного запроса к наблюдаемому устройству. Обратите внимание, что каждый процесс поллера использует свой собственный кэш.

Во всех последующих операциях получения значений проверяется только найденный индекс. Если он не изменился, значение запрашивается. Если он изменился, кэш перестраивается — каждый поллер, который встречает изменённый индекс, снова обходит и кэширует таблицу индексов SNMP.

Специальные OID

Некоторые из наиболее часто используемых SNMP OID автоматически преобразуются Zabbix в числовое представление. Например, **ifIndex** преобразуется в **1.3.6.1.2.1.2.2.1.1**, а **ifIndex.0** преобразуется в **1.3.6.1.2.1.2.2.1.1.0**.

В таблице приведен список специальных OID.

Special OID	Identifier	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Уникальное значение для каждого интерфейса.
ifDescr	1.3.6.1.2.1.2.2.1.2	Текстовая строка, содержащая сведения об интерфейсе. Эта строка должна включать имя производителя, название продукта и версию аппаратного интерфейса.
ifType	1.3.6.1.2.1.2.2.1.3	Тип интерфейса, определяемый по физическому протоколу/протоколам канального уровня, находящимся непосредственно ”ниже” сетевого уровня в стеке протоколов.
ifMtu	1.3.6.1.2.1.2.2.1.4	Размер наибольшей дейтаграммы, которую можно отправить/получить на интерфейсе, указанный в октетах.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Оценка текущей пропускной способности интерфейса в битах в секунду.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Адрес интерфейса на уровне протокола, находящемся непосредственно ‘ниже’ сетевого уровня в стеке протоколов.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Текущее административное состояние интерфейса.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Текущее рабочее состояние интерфейса.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Общее количество октетов, полученных на интерфейсе, включая символы кадрирования.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Количество одноадресных пакетов подсети, переданных протоколу более высокого уровня.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Количество неодноадресных (то есть широковещательных или многоадресных пакетов подсети), переданных протоколу более высокого уровня.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Количество входящих пакетов, которые были отброшены, хотя ошибок обнаружено не было, чтобы предотвратить их передачу протоколу более высокого уровня. Одной из возможных причин отброса такого пакета может быть освобождение буферного пространства.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Количество входящих пакетов, содержащих ошибки, из-за которых их нельзя было передать протоколу более высокого уровня.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Количество пакетов, полученных через интерфейс и отброшенных из-за неизвестного или неподдерживаемого протокола.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Общее количество октетов, переданных с интерфейса, включая символы кадрирования.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Общее количество пакетов, передача которых была запрошена протоколами более высокого уровня и которые не были адресованы многоадресному или широковещательному адресу на этом подуровне, включая отброшенные или неотправленные пакеты.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Общее количество пакетов, передача которых была запрошена протоколами более высокого уровня и которые были адресованы многоадресному или широковещательному адресу на этом подуровне, включая отброшенные или неотправленные пакеты.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Количество исходящих пакетов, которые были отброшены, хотя ошибок обнаружено не было, чтобы предотвратить их передачу. Одной из возможных причин отброса такого пакета может быть освобождение буферного пространства.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Количество исходящих пакетов, которые не удалось передать из-за ошибок.

Special OID	Identifier	Description
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Длина очереди исходящих пакетов (в пакетах).

Файлы MIB

Введение

MIB означает информационную базу управления (Management Information Base). Файлы MIB позволяют использовать текстовые представления OID (идентификаторов объектов). При мониторинге устройств SNMP с помощью Zabbix можно использовать исходные OID, но если вам удобнее работать с текстовым представлением, необходимо установить файлы MIB.

Например,

`ifHCOutOctets`

является текстовым представлением OID

`1.3.6.1.2.1.31.1.1.1.10`

Установка MIB файлов

На системах на основе Debian:

```
apt install snmp-mibs-downloader
download-mibs
```

На системах на основе RedHat:

```
dnf install net-snmp-libs
```

Подлючение MIB файлов

На системах на основе RedHat MIB файлы должны быть подключены по умолчанию. На системах на основе Debian вам нужно отредактировать файл `/etc/snmp/snmp.conf` и закомментировать строку, которая содержит `mibs`:

```
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can re-enable
# loading them by commenting out the following line.
mibs :
```

Тестирование MIB файлов

Тестирование SNMP MIB'ов можно выполнить с использованием утилиты `snmpwalk`. Если у вас эта утилита не установлена, воспользуйтесь следующими инструкциями.

На системах на основе Debian:

```
apt install snmp
```

На системах на основе RedHat:

```
dnf install net-snmp-utils
```

После чего следующая команда должна не выдавать ошибку при выполнении запроса к сетевому устройству:

```
$ snmpwalk -v 2c -c public <IP            > ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Использование MIB'ов в Zabbix

Самое главное, что следует иметь в виду, — это то, что процессы Zabbix не знают об изменениях, внесённых в MIB файлы. Поэтому после каждого изменения вам необходимо перезапустить Zabbix сервер или прокси, например:

```
systemctl restart zabbix-server
```

После выполнения этой команды изменения, сделанные в MIB файлах, вступят в силу.

Использование пользовательских MIB файлов

Имеются стандартные MIB файлы, которые поставляются с каждым GNU/Linux дистрибутивом. Но некоторые производители устройств поставляют свои собственные файлы.

Допустим, вы хотите использовать CISCO-SMI MIB файл. Следующая инструкция поможет загрузить и установить этот файл:

```
wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
mkdir -p /usr/local/share/snmp/mibs
grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/loca
cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

Теперь вы должны иметь возможность использовать его. Попытайтесь перевести имя объекта *ciscoProducts* из MIB файла в OID:

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Если вы видите ошибки вместо OID, удостоверьтесь, что все предыдущие команды не вернули никаких ошибок.

Когда перевод имени объекта заработает, вы будете готовы использовать пользовательский MIB файл. Обратите внимание, что в запросе используется MIB префикс имени (*CISCO-SMI::*). Вам потребуется его указывать, когда используете инструменты командной строки, а также Zabbix.

Не забудьте перезапустить Zabbix сервер/прокси перед использованием этого MIB файла в Zabbix.

Attention:
Имейте в виду, что у MIB файлов могут быть зависимости. То есть, одному MIB может требоваться другой. Чтобы удовлетворить эти зависимости, вам нужно установить все необходимые MIB файлы.

SNMP trap

Обзор

Получение SNMP traps — это обратный процесс по сравнению с опросом устройств с поддержкой SNMP.

В этом случае информация отправляется с устройства с поддержкой SNMP в `snmptrapd` и собирается, или "ловится", сервером Zabbix или прокси Zabbix из файла.

Обычно traps отправляются при изменении некоторого условия, а агент подключается к серверу на порт 162 (в отличие от порта 161 на стороне агента, который используется для запросов). Использование traps может выявлять некоторые кратковременные проблемы, возникающие между интервалами опроса, и которые могут быть пропущены при получении данных через опрос.

Получение SNMP traps в Zabbix предназначено для работы с `snmptrapd` и одним из механизмов передачи traps в Zabbix — либо Bash-скриптом, либо Perl-скриптом, либо SNMPTT.

Note:
Самый простой способ настроить мониторинг traps после конфигурации Zabbix — использовать решение на основе Bash-скрипта, поскольку Perl и SNMPTT часто отсутствуют в современных дистрибутивах и требуют более сложной настройки. Однако это решение использует скрипт, настроенный как `traphandle`. Для лучшей производительности в производственных системах используйте встроенное решение на Perl (либо скрипт с опцией `do perl`, либо SNMPTT).

Последовательность обработки trap:

1. `snmptrapd` получает trap.
2. `snmptrapd` передает trap скрипту-получателю (Bash, Perl) или SNMPTT.
3. Получатель разбирает, форматирует и записывает trap в файл.
4. SNMP trapper Zabbix читает и разбирает файл traps.
5. Для каждого trap Zabbix находит все элементы данных SNMP trapper с интерфейсами узлов сети, соответствующими полученному адресу trap. Обратите внимание, что при сопоставлении используется только выбранный *IP* или *DNS* в интерфейсе узла сети.

6. Для каждого найденного элемента данных trap сравнивается с regexp в `snmptrap[regexp]`. Trap устанавливается как значение **всех** совпавших элементов данных. Если совпадающий элемент данных не найден и существует элемент данных `snmptrap.fallback`, trap устанавливается как его значение.
7. Если trap не был установлен как значение ни одного элемента данных, Zabbix по умолчанию записывает в журнал несопоставленный trap. (Это настраивается параметром *Log unmatched SNMP traps* в *Administration* > *General* > *Other*.)

Замечания о переключении HA

Во время переключения ноды кластера высокой доступности (HA) Zabbix продолжит обработку после последней записи в последней отметки времени ISO 8601; если та же запись не найдена, то для определения последней позиции будет использоваться только отметка времени.

Настройка SNMP-трапов

Для этого типа элемента данных требуется следующая настройка веб-интерфейса:

Шаг 1: Создайте SNMP-интерфейс для узла сети

1. В *Сбор данных* > *Узлы сети* создайте узел сети или отредактируйте существующий.
2. В поле *Интерфейсы* выберите тип интерфейса *SNMP*.
3. Введите IP-адрес/имя DNS и номер порта.
4. Выберите *версию SNMP* из раскрывающегося списка: (*SNMPv1*, *SNMPv2* или *SNMPv3*) и добавьте учетные данные интерфейса в зависимости от выбранной версии SNMP.

Адрес каждого полученного трапа будет сравниваться с IP-адресами и адресами DNS всех SNMP-интерфейсов, чтобы найти соответствующие узлы сети.

Шаг 2: Настройте элемент данных

1. Для узла сети создайте элемент данных или отредактируйте существующий.
2. В поле *Ключ* используйте один из ключей SNMP-трапов:
   - `snmptrap[regexp]` - перехватывает все SNMP-трапы, которые соответствуют регулярному выражению, указанному в параметре `regexp`; если `regexp` не задан, перехватывает любой трап.
 В параметре поддерживаются пользовательские макросы и глобальные регулярные выражения.
 Возвращаемое значение: SNMP trap.
 Этот элемент данных можно задать только для SNMP-интерфейсов.
   - `snmptrap.fallback` - перехватывает все SNMP-трапы, которые не были перехвачены ни одним из элементов `snmptrap[]` для этого интерфейса.
 Возвращаемое значение: SNMP trap.
 Этот элемент данных можно задать только для SNMP-интерфейсов.

Note:
В настоящее время сопоставление многострочных регулярных выражений не поддерживается.

3. Установите для *Типа информации* значение *Log*, чтобы выполнялся разбор временных меток. Другие форматы, такие как *Numeric*, также допустимы, но могут потребовать пользовательский обработчик трапов.

Настройка мониторинга SNMP трапов

Настройка Zabbix server/proxy

Чтобы читать трапы, Zabbix server или proxy должны быть настроены на запуск процесса SNMP trapper и указание файла трапов, в который записывает SNMPTT или приемник трапов Bash/Perl. Для этого отредактируйте файл конфигурации (`zabbix_server.conf` или `zabbix_proxy.conf`):

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
Если используется параметр systemd `PrivateTmp`, этот файл, скорее всего, не будет работать в `/tmp`.

Настройка приемника трапов Bash

Требования: только snmptrapd.

Скрипт приемника трапов Bash можно использовать для передачи трапов в Zabbix server из snmptrapd через файл trapper. Чтобы настроить его, добавьте параметр `traphandle` в файл конфигурации snmptrapd (`snmptrapd.conf`), см. пример.

Note:
Возможно, потребуется перезапустить snmptrapd, чтобы он применил изменения в своей конфигурации.

Настройка Perl-приемника ловушек

Требования: Perl, Net-SNMP, скомпилированный с `--enable-embedded-perl` (по умолчанию включено начиная с Net-SNMP 5.4)

Perl-приемник ловушек (см. `misc/snmptrap/zabbix_trap_receiver.pl`) можно использовать для передачи ловушек на сервер Zabbix напрямую из snmptrapd. Чтобы настроить его:

- Добавьте Perl-скрипт в файл конфигурации snmptrapd (`snmptrapd.conf`), например:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Настройте приемник, например:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
Возможно, потребуется перезапустить snmptrapd, чтобы он применил изменения в своей конфигурации.

Note:
Если имя скрипта не заключено в кавычки, snmptrapd откажется запускаться, выводя сообщения, подобные следующим:


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

Настройка SNMPTT

Сначала необходимо настроить `snmptrapd` для использования SNMPTT.

Note:
Для наилучшей производительности SNMPTT следует настроить как демон, используя `snmptthandler-embedded`, чтобы передавать ему ловушки. См. инструкции по настройке SNMPTT.

Когда SNMPTT настроен на прием ловушек, настройте `snmptt.ini`:

1. Включите использование модуля Perl из пакета `NET-SNMP`:

```
net_snmp_perl_enable = 1
```

2. Записывайте ловушки в файл ловушек, который будет считываться Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. Установите формат даты и времени:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Пакет `net-snmp-perl` был удален в RHEL 8.0-8.2; снова добавлен в RHEL 8.3. Дополнительные сведения см. в разделе известные проблемы.

Теперь отформатируйте ловушки так, чтобы Zabbix мог их распознать (отредактируйте `snmptt.conf`):

1. Каждая инструкция `FORMAT` должна начинаться с `ZBXTRAP [address]`, где `[address]` будет сравниваться с IP- и DNS-адресами SNMP-интерфейсов в Zabbix. Например:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Подробнее о формате SNMP-ловушек см. ниже.

Attention:
Не используйте неизвестные ловушки — Zabbix не сможет их распознать. Неизвестные ловушки можно обработать, определив общее событие в `snmptt.conf`:


```
EVENT general .* "General event" Normal
```

Формат SNMP трапа

Все заданные получатели трапов Perl и конфигурация SNMPTT трапов должны форматировать трап следующим образом:

```
[       ] [   ,     1] ZBXTRAP [   ] [   ,     2]
```

где

- `[ ]` — штамп времени в формате `%Y-%m-%dT%H:%M:%S%z`.
- `ZBXTRAP` — заголовок, который указывает, что с этой строки начался новый трап.
- `[ ]` — IP адрес, используемый для поиска узла сети для этого трапа.

Обратите внимание, что `ZBXTRAP` и `[ ]` при обработке отрезаются из сообщения. Если трап форматируется как-то иначе, Zabbix может разобрать такие трапы неожиданным образом.

Пример трапа:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

Это приведёт к следующему виду трапа для SNMP интерфейса с IP=192.168.1.1:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

Системные требования

Note:
Рекомендуется установить файлы MIB, чтобы гарантировать, что значения элементов данных отображаются в правильном формате. Без файлов MIB могут возникнуть проблемы с форматированием, такие как отображение значений в HEX вместо UTF-8 или наоборот.

Поддержка больших файлов

Zabbix поддерживает большие файлы для файлов SNMP trapper. Максимальный размер файла, который может прочитать Zabbix, составляет 2^63 (8 EiB). Обратите внимание, что файловая система может накладывать более низкий предел на размер файла.

Ротация журнала

Zabbix не предоставляет какую-либо систему ротации журналов — это должно быть обеспечено пользователем. При ротации журнала старый файл должен быть сначала переименован и только после этого удалён, чтобы никакие трапы не пропали:

1. Zabbix открывает файл с трапами с последней известной позиции и переходит к шагу 3.
2. Zabbix проверяет, была ли выполнена ротация открытого в данный момент файла, сравнивая номера inode с заданным у файла трапов номером inode. Если открытого файла нет, Zabbix сбрасывает последнюю позицию и переходит к шагу 1.
3. Zabbix читает данные из открытого в данный момент файла и устанавливает новую позицию.
4. Обрабатываются новые данные. Если этот файл был ротирован, то он закрывается и Zabbix переходит назад к шагу 2.
5. Если не было новых данных, Zabbix засыпает на 1 секунду и возвращается к шагу 2.

Файловая система

Из-за такой реализации файла трапов Zabbix'у, чтобы различать файлы, требуется, чтобы файловая система поддерживала inode'ы (эта информация берётся из вызова `stat()`).

Примеры настройки с использованием разных версий протокола SNMP

В этом примере используются snmptrapd и скрипт-получатель Bash для передачи трапов на сервер Zabbix.

Настройка:

1. Настройте Zabbix для запуска SNMP trapper и укажите файл трапов. Добавьте в `zabbix_server.conf`:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Загрузите скрипт Bash в `/usr/sbin/zabbix_trap_handler.sh`:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/temp
```

При необходимости измените переменную `ZABBIX_TRAPS_FILE` в скрипте. Чтобы использовать значение по умолчанию, сначала создайте родительский каталог:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Добавьте следующие строки в `snmtrapd.conf` (см. рабочий пример):

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Для применения изменений конфигурации может потребоваться перезапустить snmptrapd.

4. Создайте элемент данных SNMP TEST (учтите первоначальные требования к настройке):

   Тип: *SNMP trap*
 Тип информации: *Log*
 Интерфейс узла сети: *SNMP 127.0.0.1*
 Ключ: `snmptrap["linkUp"]`
 Формат времени журнала: `yyyyMMdd.hhmmss`

Обратите внимание, что используется формат даты и времени ISO 8601.

5. Далее настроим `snmptrapd` для выбранной версии протокола SNMP и отправим тестовые трапы с помощью утилиты `snmptrap`.

SNMPv1, SNMPv2

Протоколы SNMPv1 и SNMPv2 используют аутентификацию с помощью «строки сообщества». В приведенном ниже примере мы будем использовать `secret` в качестве строки сообщества. Это же значение должно быть задано на отправителях SNMP-трапов.

Обратите внимание, что, несмотря на широкое использование в производственных средах, SNMPv2 не обеспечивает шифрование и настоящую аутентификацию отправителя. Данные передаются в открытом виде, поэтому эти версии протокола следует использовать только в защищенных средах, например в частной сети, и никогда не использовать в общедоступных сетях или сетях сторонних организаций.

SNMP версии 1 в настоящее время практически не используется, поскольку не поддерживает 64-разрядные счетчики и считается устаревшим протоколом.

Чтобы разрешить прием трапов SNMPv1 или SNMPv2, добавьте следующую строку в `snmptrapd.conf`. Замените `secret` строкой сообщества SNMP, настроенной на отправителях SNMP-трапов:

```
authCommunity log,execute,net secret
```

Теперь можно отправить тестовый трап с помощью `snmptrap`. В этом примере мы будем использовать распространенный OID «link up»:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 устраняет проблемы безопасности SNMPv1/v2 и обеспечивает аутентификацию и шифрование. Можно использовать методы аутентификации MD5 или SHA, а также DES или AES в качестве шифра.

Настройте `snmptrapd` для приема трапов SNMPv3, добавив следующие строки в `snmptrapd.conf`:

```
#####               USM SNMPv3
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
#####
authUser log,execute <securityName>
```

Например:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Обратите внимание на ключевое слово execute, которое позволяет выполнять скрипты для этой модели безопасности пользователя.

Чтобы проверить конфигурацию трапов SNMPv3, отправьте тестовый трап с помощью следующей команды:

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
Если вы хотите использовать надежные методы шифрования, такие как AES192 или AES256, используйте net-snmp начиная с версии 5.8. Возможно, потребуется перекомпилировать его с параметром configure: --enable-blumenthal-aes. Более старые версии net-snmp не поддерживают AES192/AES256. См. также: Strong Authentication or Encryption.

Проверка

В обоих примерах в /var/lib/zabbix/snmptraps/snmptraps.log вы увидите похожие строки:

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Значение элемента данных в Zabbix будет выглядеть следующим образом:

```
2024-01-30 10:04:23 2024-01-30 10:04:21

2024-01-30T10:04:21+0200 UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Пример с Perl:

```
2024-01-30T11:42:54+0200 ZBXTRAP 127.0.0.1
PDU INFO:
  receivedfrom                   UDP: [127.0.0.1]:58649->[127.0.0.1]:162
  notificationtype               TRAP
  version                        1
  community                      public
  errorstatus                    0
  transactionid                  1
  requestid                      2101882550
  messageid                      0
  errorindex                     0
VARBINDS:
  DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (457671) 1:16:16.71
  SNMPv2-MIB::snmpTrapOID.0      type=6  value=OID: IF-MIB::linkUp.0
```

Смотрите также

- Статья в Zabbix блоге о SNMP трапах [en]
- Настройка snmptrapd (официальная документация net-snmp) [en]
- Настройка snmptrapd на получение SNMPv3 уведомлений (официальная документация net-snmp) [en]

Внутренний Zabbix

Обзор

Внутренние проверки позволяют отслеживать внутренние процессы Zabbix. Иными словами, можно отслеживать происходящее с сервером Zabbix или прокси Zabbix.

Внутренние проверки выполняются:

- на сервере Zabbix — если мониторинг узла сети выполняется сервером
- на прокси Zabbix — если мониторинг узла сети выполняется прокси

Внутренние проверки обрабатываются сервером или прокси независимо от состояния обслуживания узла сети.

Чтобы использовать этот элемент данных, выберите тип элемента данных **Zabbix internal**.

Note:
Внутренние проверки обрабатываются опросчиками Zabbix.

Производительность

Использование некоторых внутренних элементов данных может отрицательно повлиять на производительность. Речь идёт о следующих элементах данных:

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue,,]
- zabbix[requiredperformance]
- zabbix[stats,,,queue,,]
- zabbix[triggers]

Также это влияет на разделы веб-интерфейса Информация о системе (System information) и Очередь (Queue).

Поддерживаемые проверки

Ключи элементов данных перечислены без настраиваемых параметров и дополнительной информации.
Нажмите на ключ элемента данных, чтобы просмотреть полную информацию.

Ключ элемента данных	Описание
zabbix[apm,]	Статистика телеметрических данных, полученных сборщиком APM.
zabbix[boottime]	Время запуска процесса сервера Zabbix или прокси Zabbix в секундах.
zabbix[cluster,discovery,nodes]	Обнаруживает узлы кластера высокой доступности.
zabbix[connector_queue]	Количество значений, помещенных в очередь коннектора.
zabbix[discovery_queue]	Количество сетевых проверок, помещенных в очередь обнаружения.
zabbix[host„items]	Количество включенных элементов данных (поддерживаемых и неподдерживаемых) на узле сети.
zabbix[host„items_unsupported]	Количество включенных неподдерживаемых элементов данных на узле сети.
zabbix[host„maintenance]	Текущий статус обслуживания узла сети.
zabbix[host,active_agent,available]	Доступность проверок активным агентом на узле сети.
zabbix[host,discovery,interfaces]	Подробная информация обо всех настроенных интерфейсах узла сети в веб-интерфейсе Zabbix.
zabbix[host„available]	Доступность основного интерфейса для определенного типа проверок на узле сети.
zabbix[hosts]	Количество отслеживаемых узлов сети.
zabbix[items]	Количество включенных элементов данных (поддерживаемых и неподдерживаемых).
zabbix[items_unsupported]	Количество неподдерживаемых элементов данных.
zabbix[java„]	Информация о Java gateway Zabbix.
zabbix[lld_queue]	Количество значений, помещенных в очередь обработки низкоуровневого обнаружения.
zabbix[preprocessing]	Статистика значений, полученных менеджером предварительной обработки.
zabbix[preprocessing_queue]	Количество значений, помещенных в очередь предварительной обработки.
zabbix[process„„]	Процент времени, затраченного определенным процессом Zabbix или группой процессов (идентифицируемых с помощью <type> и <mode>) в состоянии <state>.
zabbix[proxy„]	Информация о прокси Zabbix.
zabbix[proxy,discovery]	Список прокси Zabbix.
zabbix[proxy group„available]	Количество прокси в сети в группе прокси.
zabbix[proxy group„pavailable]	Процент прокси в сети в группе прокси.
zabbix[proxy group„proxies]	Список прокси Zabbix в группе прокси.
zabbix[proxy group„state]	Состояние группы прокси.
zabbix[proxy group,discovery]	Возвращает список групп прокси с данными конфигурации и данными в реальном времени.
zabbix[proxy_buffer,buffer,]	Возвращает статистику использования буфера памяти прокси.

Ключ элемента данных	Описание
zabbix[proxy_buffer,state,changes]	Возвращает количество изменений состояния между режимами дискового буфера и буфера памяти с момента запуска.
zabbix[proxy_buffer,state,current]	Возвращает текущее рабочее состояние, в котором сохраняются новые данные.
zabbix[proxy_history]	Количество значений в таблице истории прокси, ожидающих отправки на сервер.
zabbix[queue,,]	Количество отслеживаемых элементов данных в очереди, задержанных как минимум на <from> секунд, но менее чем на <to> секунд.
zabbix[rcache,,]	Статистика доступности кэша конфигурации Zabbix.
zabbix[requiredperformance]	Требуемая производительность сервера Zabbix или прокси в ожидаемом количестве новых значений в секунду.
zabbix[stats,,]	Возвращает внутренние метрики сервера Zabbix или прокси. Если указаны <ip> и <port>, метрики извлекаются из удаленного экземпляра; в противном случае — из локального экземпляра.
zabbix[stats,,,queue,,]	Возвращает внутренние метрики очереди сервера Zabbix или прокси. Если указаны <ip> и <port>, метрики извлекаются из удаленного экземпляра; в противном случае — из локального экземпляра.
zabbix[tcache,,]	Статистика эффективности кэша функции трендов Zabbix.
zabbix[triggers]	Количество включенных триггеров в базе данных Zabbix, у которых все элементы данных включены на включенных узлах сети.
zabbix[uptime]	Время работы процесса сервера Zabbix или прокси в секундах.
zabbix[vcache,buffer,]	Статистика доступности кэша значений Zabbix.
zabbix[vcache,cache,]	Статистика эффективности кэша значений Zabbix.
zabbix[version]	Версия сервера Zabbix или прокси.
zabbix[vmware,buffer,]	Статистика доступности кэша vmware Zabbix.
zabbix[vps,written]	Общее количество значений истории, записанных в базу данных.
zabbix[wcache,,]	Статистика и доступность кэша записи Zabbix.

Подробности ключа элемента данных

- Параметры без угловых скобок являются обязательными и должны использоваться *как есть* (например, «host» и «available» в `zabbix[host,< >,available]`).
- Параметры с угловыми скобками **< >** должны заменяться корректными значениями. Если параметр имеет значение по умолчанию, его можно опустить.
- Значения элементов данных и параметров элементов данных с пометкой «не поддерживаются на прокси» можно собирать только на узлах сети, которые наблюдаются через сервер. И наоборот, значения «не поддерживается на сервере» можно собирать, только если узел сети наблюдается через прокси.

zabbix[apm,<type>]

 Статистика телеметрических данных, полученных сборщиком APM:

- *requests* - количество запросов, полученных сборщиком APM (счетчик):
  - *accepted* - количество запросов, принятых для обработки;
  - *dropped* - количество отброшенных запросов;
- *rows* - количество строк телеметрических данных, записанных в источник данных, для каждого типа данных (счетчик).

Возвращаемое значение: *объект JSON*.

Параметры:

- **type** - *traces*, *metrics* или *logs*. Если параметр не указан, возвращается статистика для всех типов.

Комментарии:

- Этот элемент данных не поддерживается на сервере Zabbix.

Пример возвращаемых значений для `zabbix[apm]`:

```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
```

```
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```

zabbix[boottime]

 Время запуска процесса Zabbix сервера или Zabbix прокси в секундах.
 Возвращаемое значение: *Целое число*.

zabbix[cep]

 Complex event processing statistics.
 Return value: *JSON*.

zabbix[cluster,discovery,nodes]

 Обнаруживает узлы сети кластера высокой доступности.
 Возвращаемое значение: *объект JSON*.

Комментарии:

- Этот элемент данных можно использовать в обнаружении низкого уровня.
- Возвращаемые поля в основном соответствуют свойствам объекта узла сети высокой доступности.
- Дополнительные поля:
  - `db_timestamp` - текущее время сервера (метка времени Unix), включенное в ответ
  - `lastaccess_age` - количество секунд с момента последнего сигнала узла (db_timestamp - lastaccess)
  - `status` - состояние узла сети:
    * 0 - резервный
    * 1 - остановлен вручную
    * 2 - недоступен
    * 3 - активен

Пример возвращаемых значений:

```
[
  {
    "id": "ckvupihk70001z8mkpw5cg0u3",
    "name": "zabbix-prod-01",
    "status": 3,
    "address": "10.0.4.12:10051",
    "port": 10051,
    "lastaccess": 1756115995,
    "db_timestamp": 1756116000,
    "lastaccess_age": 5
  },
  {
    "id": "ckvx2a9k70004b1nq2hz9d7f",
    "name": "zabbix-standby-02",
    "status": 0,
    "address": "10.0.4.13:10051",
    "port": 10051,
    "lastaccess": 1756115550,
    "db_timestamp": 1756116000,
    "lastaccess_age": 450
  },
  {
    "id": "ckw0bq3l70007y4r1a0m5kz8",
    "name": "zabbix-backup-eu1",
    "status": 1,
    "address": "backup.example.com:10051",
    "port": 10051,
    "lastaccess": 1756105080,
    "db_timestamp": 1756116000,
    "lastaccess_age": 10920
  }
```

```
]
```

zabbix[connector_queue]

 Количество значений, поставленных в очередь коннектора..
 Возвращаемое значение: *Целое число*.

zabbix[discovery_queue]

 Количество сетевых проверок, поставленных в очередь обнаружения.
 Возвращаемое значение: *Целое число*.

zabbix[host,,items]

 Количество активированных элементов данных (поддерживаемых и неподдерживаемых) у узла сети.
 Возвращаемое значение: *Целое число*.

zabbix[host,,items_unsupported]

 Количество активированных неподдерживаемых элементов данных у узла сети.
 Возвращаемое значение: *Целое число*.

zabbix[host,,maintenance]

 Текущее состояние обслуживания узла сети.
 Возвращаемое значение: *0* — нормальное состояние; *1* — обслуживание со сбором данных; *2* — обслуживание без сбора данных.

Комментарии:

- Этот элемент данных всегда обрабатывается Zabbix сервером вне зависимости от настроек узла сети (наблюдается через сервер или прокси). Прокси не получает этот элемент данных при получении своей конфигурации.
- Второй параметр должен быть пустым и зарезервирован для использования в будущем.

zabbix[host,active_agent,available]

 Доступность активных проверок агента на узле сети.
 Возвращаемые значения: *0* — неизвестно; *1* — доступно; *2* — недоступно.

zabbix[host,discovery,interfaces]

 Сведения по всем настроенным интерфейсам узла сети в веб-интерфейсе Zabbix.
 Возвращаемое значение: *Объект JSON*.

Комментарии:

- Этот элемент данных можно использовать в низкоуровневом обнаружении.
- Этот элемент данных не поддерживается на Zabbix прокси.

zabbix[host,<тип>,available]

 Доступность основного интерфейса определённого типа проверок на узле сети.
 Возвращаемые значения: *0* — недоступен; *1* — доступен; *2* — неизвестно.

Параметры:

- **тип** — *agent*, *snmp*, *ipmi* или *jmx*.

Комментарии:

- Значение элемента данных вычисляется согласно параметрам конфигурации доступности/недоступности касательно узлов сети.

zabbix[hosts]

 Количество наблюдаемых узлов сети.
 Возвращаемое значение: *Целое число*.

zabbix[items]

 Количество активированных элементов данных (поддерживаемых и неподдерживаемых).
 Возвращаемое значение: *Целое число*.

zabbix[items_unsupported]

 Количество неподдерживаемых элементов данных.
 Возвращаемое значение: *Целое число*.

zabbix[java,,<param>]

 Информация о Zabbix Java gateway.
 Возвращаемые значения: *1* - если <param> равно *ping*; *версия Java gateway* - если <param> равно *version* (например: ”8.0.0”).

Параметры:

- **param** - *ping* или *version*.

Комментарии:

- Этот элемент данных можно использовать для проверки доступности Java gateway с помощью функции триггера `nodata()`.
- Второй параметр должен быть пустым и зарезервирован для будущего использования.

zabbix[lld_queue]

 Количество значений, помещённых в очередь обработки низкоуровневых обнаружений.
 Возвращаемое значение: *Целое число*.

Комментарии:

- Этот элемент данных можно использовать для мониторинга размера очереди обработки низкоуровневых обнаружений.

zabbix[preprocessing]

 Статистика значений, полученных менеджером предварительной обработки:

- *queued* - количество и размер значений в очереди, требующих предварительной обработки (счетчик)
- *direct* - количество и размер значений в очереди, не требующих предварительной обработки (счетчик)
- *queue* - количество значений, добавленных в очередь предварительной обработки (то же, что и `zabbix[preprocessing_queue`

Возвращаемое значение: *JSON*.

Пример возвращаемых значений:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 Количество значений, помещённых в очередь предварительной обработки.
 Возвращаемое значение: *Целое число*.

Комментарии:

- Этот элемент данных можно использовать для мониторинга размера очереди предварительной обработки.

zabbix[process,<type>,<mode>,<state>]

 Процент времени, которое определённый процесс Zabbix или группа процессов (определяемые параметрами `<type>` и `<mode>`) провели в состоянии `<state>`. Рассчитывается только за последнюю минуту.
 Возвращаемое значение: *Float*.

Параметры:

- **type**:
  - для процессов сервера: *agent poller*, *alert manager*, *alert syncer*, *alerter*, *availability manager*, *browser poller*, *configuration syncer*, *configuration syncer worker*, *connector manager*, *connector worker*, *discovery manager*, *discovery worker*, *escalator*, *ha manager*, *history poller*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *lld manager*, *lld worker*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *proxy group manager*, *proxy poller*, *self-monitoring*, *service manager*, *snmp poller*, *snmp trapper*, *task manager*, *timer*, *trapper*, *trigger housekeeper*, *unreachable poller*, *vmware collector*;
  - для процессов прокси: *agent poller*, *apm manager*, *apm worker*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;

- **mode**:
  - *avg* — среднее значение для всех процессов указанного типа (по умолчанию);
  - *count* — возвращает количество ответвлений для указанного типа процесса; <state> указывать не следует;
  - *max* — максимальное значение;
  - *min* — минимальное значение;
  - *<process number>* — номер процесса (от 1 до количества предварительно созданных экземпляров; например, если запущены 4 процесса *trapper*, значение находится в диапазоне от 1 до 4);
- **state**:
  - *busy* — процесс находится в занятом состоянии, например обрабатывает запрос (по умолчанию);
  - *idle* — процесс находится в состоянии ожидания и ничего не делает.

Комментарии:

- Если <mode> — это номер процесса Zabbix, который не запущен (например, если запущено 5 процессов *poller*, а для <mode> указано значение 6), такой элемент данных перейдёт в состояние «не поддерживается».
- Минимальное и максимальное значения относятся к проценту использования одного процесса. Таким образом, если в группе из 3 процессов *poller* процент использования процессов составляет 2, 18 и 66, *min* вернёт 2, а *max* — 66.
- Процессы сообщают о выполняемых действиях в общей памяти, а процесс самоконтроля обобщает эти данные каждую секунду. Изменения состояния (busy/idle) регистрируются при изменении — поэтому процесс, перешедший в состояние busy, регистрирует это состояние и не изменяет его и не обновляет до перехода в состояние idle. Это гарантирует, что даже полностью зависшие процессы будут корректно зарегистрированы как занятые на 100%.
- В настоящее время «busy» означает «не находится в состоянии сна», однако в будущем могут быть добавлены дополнительные состояния — ожидание блокировок, выполнение запросов к базе данных и т. д. Обратите внимание, что асинхронные процессы *poller* считаются занятыми, если они достигли ограничения, заданного параметром конфигурации MaxConcurrentChecksPerPoller для сервера/прокси.
- В Linux и большинстве других систем разрешение составляет 1/100 секунды.

Примеры:

```
zabbix[process,poller,avg,busy] #       ,           poller         -
zabbix[process,"icmp pinger",max,busy] #          ,              ICMP pinger         -
zabbix[process,"history syncer",2,busy] #   ,          history syncer       2         -
zabbix[process,trapper,count] #                           trapper
```

zabbix[proxy,<name>,<param>]

 Информация о прокси Zabbix.
 Возвращаемое значение: *Integer*.

Параметры:

- **name** — имя прокси;
- **param**:
  - *lastaccess* — временная метка последнего сообщения heartbeat, полученного от прокси;
  - *delay* — время, в течение которого собранные значения остаются неотправленными; рассчитывается как «задержка прокси» + («текущее время сервера» - «lastaccess прокси»), где «задержка прокси» — это разница между текущим временем прокси и временной меткой самого старого неотправленного значения на прокси.

Комментарии:

- Этот элемент данных не поддерживается на узлах сети, мониторинг которых выполняется прокси или группой прокси.
- Для проверки доступности прокси можно использовать функцию fuzzytime().

Пример:

```
zabbix[proxy,"Germany",lastaccess] #                      heartbeat,              "Germany"
```

zabbix[proxy,discovery]

 Список прокси Zabbix с указанием имени, режима, шифрования, сжатия, версии, времени последнего обнаружения, количества узлов сети, количества элементов данных, требуемой производительности в секунду (vps), статуса версии (текущая/устаревшая/неподдерживаемая), тайм-аутов для типов элементов данных, имени группы прокси (если прокси входит в группу), состояния (неизвестное/не в сети/в сети).
 Возвращаемое значение: *объект JSON*.

Комментарий:

- Этот элемент данных не поддерживается на узлах сети, которые отслеживаются прокси или группой прокси.

Примеры возвращаемых значений:

```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```


zabbix[proxy group,<name>,available]

 Количество прокси в группе прокси, находящихся в сети.
 Возвращаемое значение: *Integer*.

Параметры:

- **name** - имя группы прокси.

Комментарий:

- Этот элемент данных не поддерживается на узлах сети, мониторинг которых выполняется прокси или группой прокси.

zabbix[proxy group,<name>,pavailable]

 Процент доступных прокси в группе прокси.
 Возвращаемое значение: *Float*.

Параметры:

- **name** - имя группы прокси.

Комментарий:

- Этот элемент данных не поддерживается для узлов сети, мониторинг которых выполняется прокси или группой прокси.

zabbix[proxy group,<name>,proxies]

 Список прокси Zabbix в группе прокси с указанием имени, режима, шифрования, сжатия, версии, времени последнего обнаружения, количества узлов сети, количества элементов данных, требуемого количества значений в секунду (vps), статуса версии (текущая/устаревшая/неподдерживаемая), времени ожидания, имени группы прокси и состояния (неизвестное/не в сети/в сети).
 Возвращаемое значение: *JSON*.

Параметры:

- **name** - имя группы прокси.

Комментарий:

- Этот элемент данных не поддерживается на узлах сети, которые отслеживаются прокси или группой прокси.

zabbix[proxy group,<name>,state]

 Состояние группы прокси.
 Возвращаемое значение: *0* — неизвестно; *1* — не в сети; *2* — восстановление; *3* — в сети; *4* — ухудшение состояния.

Параметры:

- **name** — имя группы прокси.

Комментарий:

- Этот элемент данных не поддерживается для узлов сети, мониторинг которых выполняется прокси или группой прокси.

zabbix[proxy group,discovery]

 Возвращает список групп прокси с данными конфигурации и данными в реальном времени. Данные конфигурации включают имя группы прокси, задержку переключения при отказе и минимальное количество прокси в сети. Данные в реальном времени включают состояние группы прокси (подробности см. в комментариях), количество прокси в сети и процент прокси в сети.
 Возвращаемое значение: *JSON*.

Комментарии:

- Этот элемент данных не поддерживается на узлах сети, мониторинг которых выполняется прокси или группой прокси.
- Этот элемент данных не возвращает прокси, не входящие в группы.
- Если указано недопустимое значение для `failover_delay` или `min_online`, сообщается специальное значение −1. Недопустимые значения могут возникнуть, если для конфигурации используются макросы, которые невозможно развернуть в допустимое значение.
- Состояние группы прокси сообщается в виде целого числа: *0* - неизвестно; *1* - не в сети; *2* - восстановление; *3* - в сети; *4* - ухудшение состояния.

Пример возвращаемых значений:

```
{
    "groups": [
       { "name": "Riga", "failover_delay": 60, "min_online": 1 },
       { "name": "Tokyo", "failover_delay": 60, "min_online": 2 },
       { "name": "Porto Alegre", "failover_delay": 60, "min_online": 3 }
    ],
    "details": {
        "Riga": { "state": 3, "available": 10, "pavailable": 20 },
        "Tokyo": { "state": 3, "available": 10, "pavailable": 20 },
        "Porto Alegre": { "state": 1, "available": 0, "pavailable": 0 }
    }
}
```

zabbix[proxy_buffer,buffer,<mode>]

 Статистика использования буфера памяти прокси.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процентного значения).

Параметры:

- **mode**:
  - *total* - общий размер буфера (можно использовать для проверки, включен ли буфер памяти);
  - *free* - размер свободного буфера;
  - *pfree* - процент свободного буфера;
  - *used* - размер используемого буфера;
  - *pused* - процент используемого буфера.

Комментарии:

- Возвращает ошибку «Буфер памяти прокси отключен», если буфер памяти отключен;
- Этот элемент данных не поддерживается на сервере Zabbix.

zabbix[proxy_buffer,state,changes]

 Возвращает количество изменений состояния между режимами буфера диск/память с момента запуска.
 Возвращаемое значение: *Целое число*; *0* - буфер памяти отключён.

Комментарии:

- Частые изменения состояния указывают на то, что необходимо увеличить размер или возраст буфера памяти.
- Если состояние буфера памяти отслеживается нечасто (например, раз в минуту), то буфер может изменить своё состояние, не отметив этого.

zabbix[proxy_buffer,state,current]

 Возвращает текущее рабочее состояние, в котором сохраняются новые данные.
 Возвращаемые значения: *0* — диск; *1* — память.

Комментарий:

- 0 также возвращается, если буфер в памяти отключен.

zabbix[proxy_history]

 Количество значений в таблице истории прокси, ожидающих отправки на сервер.
 Возвращаемые значения: *Integer*.

Комментарий:

- Этот элемент данных не поддерживается на сервере Zabbix.

zabbix[queue,<from>,<to>]

 Количество контролируемых элементов данных в очереди, задержка которых составляет не менее `<from>` секунд, но менее `<to>` секунд.
 Возвращаемое значение: *Integer*.

Параметры:

- **from** - задержка не менее (по умолчанию 6 секунд);
- **to** - задержка не более (по умолчанию бесконечность).

Комментарии:

- В параметрах поддерживаются суффиксы времени (`s,m,h,d,w`).

Attention:
Прокси Zabbix не учитывает периоды обслуживания; см. Расчет очередей во время обслуживания для подробностей.

zabbix[rcache,<cache>,<mode>]

 Статистика доступности кэша конфигурации Zabbix.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процента).

Параметры:

- **cache** - *buffer*;
- **mode**:
  - *total* - общий размер буфера;
  - *free* - размер свободного буфера;
  - *pfree* - процент свободного буфера;
  - *used* - размер используемого буфера;
  - *pused* - процент используемого буфера.

zabbix[requiredperformance]

 Требуемая производительность сервера Zabbix или прокси Zabbix, в ожидаемом количестве новых значений в секунду.
 Возвращаемое значение: *Float*.

Комментарии:

- Примерно соответствует пункту ”Required server performance, new values per second” в *Reports* > *System information*.

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

 Возвращает внутренние метрики сервера или прокси Zabbix. Если указаны `<ip>` и `<port>`, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.
 Возвращаемые значения: *JSON object*.

Параметры:

- **ip** - список IP/DNS/масок сети серверов/прокси, к которым выполняется удаленный запрос (по умолчанию 127.0.0.1);
- **port** - порт сервера/прокси, к которому выполняется удаленный запрос (по умолчанию 10051).

Комментарии:

- Запрос stats будет принят только с адресов, перечисленных в параметре `StatsAllowedIP` server/proxy на целевом экземпляре.
- Этот элемент данных возвращает выбранный набор внутренних метрик. Подробности см. в разделе Удаленный мониторинг статистики Zabbix.

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

 Возвращает метрики внутренней очереди (см. `zabbix[queue,<from>,<to>]`) сервера Zabbix или прокси. Если указаны <ip> и <port>, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.
 Возвращаемые значения: *JSON object*.

Параметры:

- **ip** - список IP/DNS/сетевых масок серверов/прокси, к которым будет выполнен удаленный запрос (по умолчанию `127.0.0.1`);
- **port** - порт сервера/прокси, к которому будет выполнен удаленный запрос (по умолчанию 10051);
- **from** - задержка не менее чем на (по умолчанию 6 секунд);
- **to** - задержка не более чем на (по умолчанию бесконечность).

Комментарии:

- Запрос stats будет принят только с адресов, перечисленных в параметре `StatsAllowedIP` server/proxy на целевом экземпляре.
- Этот элемент данных возвращает выбранный набор внутренних метрик. Подробности см. в разделе Удаленный мониторинг статистики Zabbix.

> **Attention:**
> Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Вычисление очередей во время обслуживания.

zabbix[tcache,cache,<parameter>]

 Статистика эффективности кэша функции трендов Zabbix.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процента).

Параметры:

- **parameter**:
  - *all* - общее количество запросов к кэшу (по умолчанию);
  - *hits* - попадания в кэш;
  - *phits* - процент попаданий в кэш;
  - *misses* - промахи кэша;
  - *pmisses* - процент промахов кэша;
  - *items* - количество кэшированных элементов;
  - *requests* - количество кэшированных запросов;
  - *pitems* - процент кэшированных элементов от кэшированных элементов + запросов. Низкий процент, скорее всего, означает, что размер кэша можно уменьшить.
  - *paccessed* - процент обращений к записям за последние 24 часа.

Комментарии:

- Этот элемент данных не поддерживается на прокси Zabbix.

zabbix[triggers]

 Количество активированных триггеров в базе данных Zabbix, со всеми активированными элементами данных, которые упомянуты в выражениях триггеров на активированных узлах сети.
 Возвращаемое значение: *Целое число*.

Комментарии:

- Этот элемент данных не поддерживается на Zabbix прокси.

zabbix[uptime]

 Время непрерывной работы процесса Zabbix сервера или прокси в секундах.
 Возвращаемое значение: *Целое число*.

zabbix[vcache,buffer,<mode>]

 Статистика доступности кэша значений Zabbix.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процента).

Параметры:

- **mode**:
  - *total* - общий размер буфера;
  - *free* - размер свободного буфера;
  - *pfree* - процент свободного буфера;
  - *used* - размер используемого буфера;
  - *pused* - процент используемого буфера.

Комментарии:

- Этот элемент данных не поддерживается на прокси Zabbix.

zabbix[vcache,cache,<parameter>]

 Статистика эффективности кэша значений Zabbix.
 Возвращаемое значение: *Integer*. Если для <parameter> задано значение *mode*, возвращаемое значение: *0* - нормальный режим; *1* - режим низкой памяти.

Параметры:

- **parameter**:
  - *requests* - общее количество запросов;
  - *hits* - количество попаданий в кэш (значения истории, взятые из кэша);
  - *misses* - количество промахов кэша (значения истории, взятые из базы данных);
  - *mode* - режим работы кэша значений.

Комментарии:

- После включения режима низкой памяти кэш значений будет оставаться в этом состоянии в течение 24 часов, даже если проблема, вызвавшая этот режим, будет устранена раньше.
- Вы можете использовать этот ключ с шагом предварительной обработки *Change per second*, чтобы получать статистику значений в секунду.
- Этот элемент данных не поддерживается на прокси Zabbix.

zabbixversion

 Версия сервера или прокси Zabbix.
 Возвращаемое значение: *String*. Например: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Статистика доступности кэша Zabbix vmware.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процента).

Параметры:

- **mode**:
  - *total* - общий размер буфера;
  - *free* - размер свободного буфера;
  - *pfree* - процент свободного буфера;
  - *used* - размер используемого буфера;
  - *pused* - процент используемого буфера.

zabbix[vps,written]

 Общее количество значений истории, записанных в базу данных.
 Возвращаемое значение: *Целое число*.

zabbix[wcache,<cache>,<mode>]

 Статистика и доступность write cache Zabbix.
 Возвращаемые значения: *Integer* (для количества/размера); *Float* (для процента).

Параметры:

- **cache** - *values*, *history*, *index* или *trend*;
- **mode**:
  - с параметром кэша *values*:
    - *all* - общее количество значений, обработанных сервером/прокси Zabbix, за исключением неподдерживаемых элементов данных (счетчик) (по умолчанию);
    - *float* - количество обработанных значений с плавающей точкой (счетчик);
    - *uint* - количество обработанных беззнаковых целых значений (счетчик);
    - *str* - количество обработанных символьных/строковых значений (счетчик);
    - *log* - количество обработанных лог-значений (счетчик);
    - *text* - количество обработанных текстовых значений (счетчик);
    - *bin* - количество обработанных двоичных значений (счетчик);
    - *json* - количество обработанных JSON-значений (счетчик);

    - *not supported* - количество случаев, когда обработка элемента данных приводила к тому, что элемент данных становился неподдерживаемым или оставался в этом состоянии (счетчик);
  - с параметрами кэша *history*, *index*, *trend*:
    - *pfree* - процент свободного буфера (по умолчанию);
    - *total* - общий размер буфера;
    - *free* - размер свободного буфера;
    - *used* - размер используемого буфера;
    - *pused* - процент используемого буфера.

Комментарии:

- Параметр кэша `trend` не поддерживается в Zabbix proxy.
- History cache используется для хранения значений элементов данных. Низкое значение указывает на проблемы с производительностью на стороне базы данных.
- History index cache используется для индексирования значений, хранящихся в history cache.
- После того как history cache заполнится и затем будет очищен, history index cache все равно будет хранить часть данных. Такое поведение ожидаемо и помогает системе работать эффективнее, избегая дополнительной обработки, необходимой для постоянного изменения размера памяти.
- Trend cache хранит агрегированные данные за текущий час для всех элементов данных, которые получают данные.
- Вы можете использовать ключ `zabbix[wcache,values]` со шагом предварительной обработки *Change per second*, чтобы получать статистику значений в секунду.

Zabbix trapper

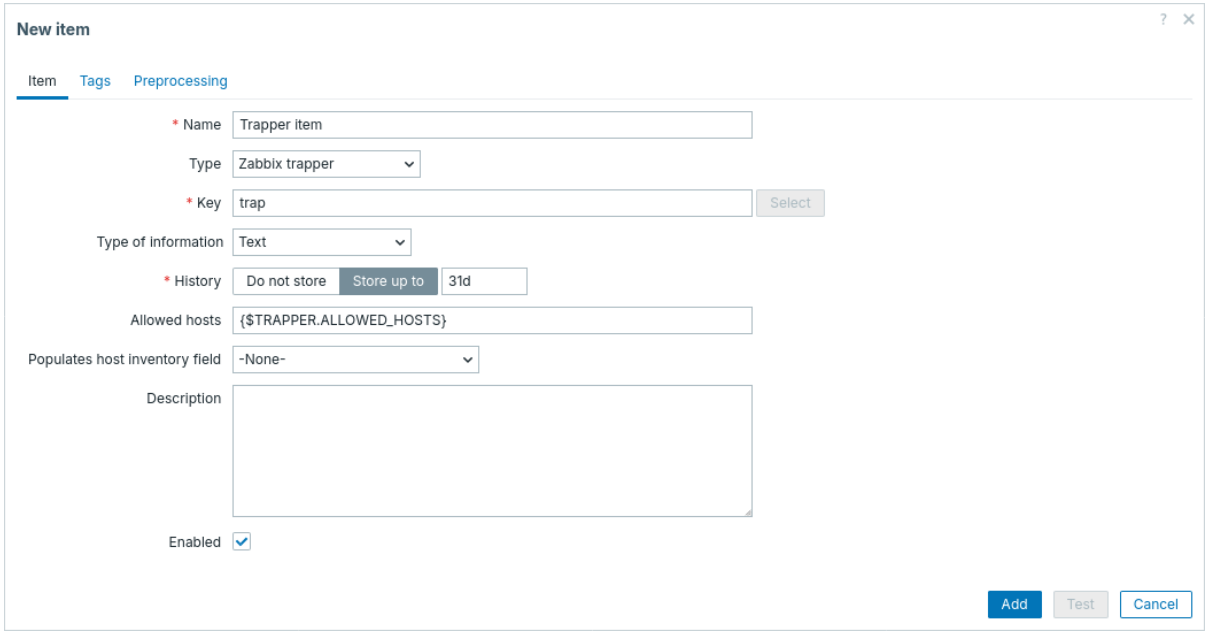
Обзор

Элементы данных типа trapper принимают входящие данные вместо того, чтобы запрашивать их. Это полезно для любых данных, которые вы хотите отправлять в Zabbix.

Конфигурация

Чтобы настроить элемент данных типа trapper:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Нажмите *Элементы данных* в строке узла сети.
3. Нажмите *Создать элемент данных*.
4. Введите параметры элемента данных в форме настройки.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, которые требуют специальной информации для элементов данных типа trapper:

Type	Выберите ”Zabbix trapper”.
Key	Введите ключ, который будет использоваться для распознавания элемента данных при отправке данных на сервер Zabbix.

Type of information	Выберите тип информации, который будет соответствовать формату отправляемых данных.
Allowed hosts	Список IP-адресов, разделенных запятыми (при необходимости в нотации CIDR), или DNS-имен. Значение по умолчанию — user macro {$TRAPPER.ALLOWED_HOSTS}. По умолчанию {$TRAPPER.ALLOWED_HOSTS} равно 127.0.0.1,::1. Вы можете либо ввести адреса напрямую в поле, либо изменить значение макроса. Если оставить поле пустым, входящие соединения будут отклоняться со всех узлов сети. Если значение указано, входящие соединения будут приниматься только от перечисленных здесь узлов сети. Если включена поддержка IPv6, то '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' считаются эквивалентными, а '::/0' разрешит любой IPv4- или IPv6-адрес. '0.0.0.0/0' можно использовать, чтобы разрешить любой IPv4-адрес. Обратите внимание, что "IPv4-compatible IPv6 addresses" (префикс 0000::/96) поддерживаются, но считаются устаревшими согласно RFC4291. Пример: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} Поддерживаются пробелы, пользовательские макросы и макросы узла сети {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Перед отправкой значений после сохранения элемента данных может потребоваться подождать до 60 секунд, пока сервер Zabbix не применит изменения из обновленного кэша конфигурации.

Передача данных

Передача данных на сервер Zabbix или прокси возможна с помощью утилиты Zabbix sender или протокола Zabbix sender. Передача данных на сервер Zabbix также возможна с помощью метода API `history.push`.

Zabbix sender

Для отправки данных на сервер Zabbix или прокси с помощью утилиты Zabbix sender можно выполнить следующую команду, чтобы отправить значение "test value":

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Для отправки значения "test value" используются следующие параметры команды:

- `-z` — указать IP-адрес сервера Zabbix
- `-p` — указать номер порта сервера Zabbix (по умолчанию 10051)
- `-s` — указать узел сети (обязательно используйте техническое имя вместо отображаемого имени узла сети)
- `-k` — указать ключ элемента данных настроенного в элементе данных типа trapper
- `-o` — указать отправляемое значение

Attention:
Процесс Zabbix trapper не разворачивает макросы, используемые в ключе элемента данных, чтобы проверить наличие соответствующего ключа элемента данных для целевого узла сети.

Дополнительные сведения о взаимодействии между Zabbix sender и сервером Zabbix или прокси см. в разделе *Протокол Zabbix sender*.

history.push

Для отправки данных на сервер Zabbix с помощью метода API `history.push` вы можете сделать следующий запрос HTTP POST, содержащий некоторые тестовые значения:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

Если запрос корректный, то ответ, возвращаемый API, может выглядеть следующим образом:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

Ошибки в данных ответа указывают на то, что отправка данных для определённых элементов данных не прошла проверку сервером Zabbix. Это может произойти по следующим причинам:

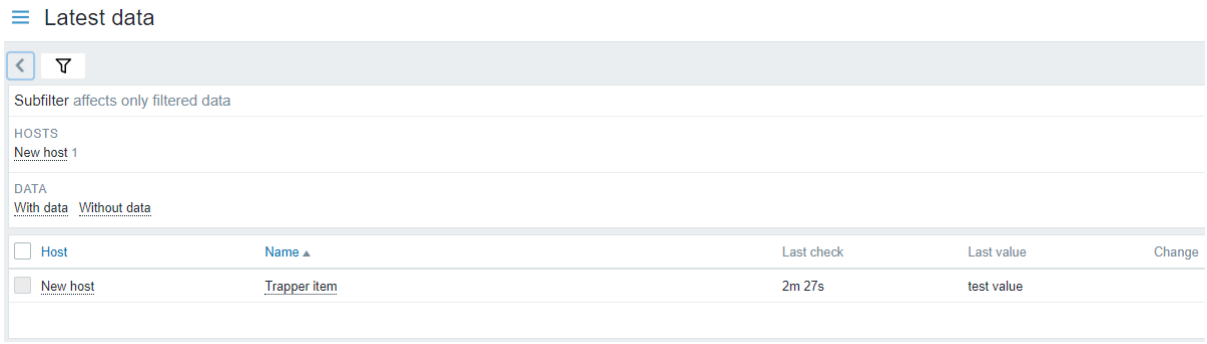
- пользователь, отправляющий данные, не имеет права *чтения* на узел сети элемента данных;
- узел отключён или находится в состоянии обслуживания без сбора данных;
- элемент данных не существует или ещё не включён в кэш конфигурации сервера;
- элемент данных отключён или его тип отличается от Zabbix траппер или HTTP агент (с включённой опцией «Активировать трапы»);
- IP или DNS пользователя не указаны в списке *Разрешённые хосты* элемента данных;
- другой элемент данных имеет значение с дублирующейся временной меткой на уровне наносекунд.

Отсутствие ошибок указывает на то, что отправленные значения были приняты для обработки, которая включает предобработку (если таковая имеется), обработку триггера и сохранение в базе данных. Обратите внимание, что обработка принятого значения также может завершиться ошибкой (например, во время предобработки), в результате чего значение будет отброшено.

Более подробную информацию о работе с API Zabbix смотрите в API.

Отображение данных

После отправки данных вы можете перейти в *Мониторинг* > *Последние данные*, чтобы увидеть результат:



Note:
Если отправлено одно числовое значение, график данных будет показывать горизонтальную линию слева и справа от временной точки значения.

Внешняя проверка

Обзор

Внешняя проверка — это проверка, выполняемая сервером Zabbix путем запуска shell-скрипта или бинарного файла. Однако, если узлы сети контролируются через прокси Zabbix, внешние проверки выполняются прокси.

Для внешних проверок не требуется, чтобы на контролируемом узле сети был запущен какой-либо агент.

Синтаксис ключа элемента данных:

`script[<parameter1>,<parameter2>,...]`

Где:

ARGUMENT	DEFINITION
script	Имя shell-скрипта или бинарного файла.
parameter(s)	Необязательные параметры командной строки.

Если вы не хотите передавать скрипту какие-либо параметры, можно использовать:

```
script[] or
script
```

Сервер Zabbix или прокси выполнит поиск в каталоге, указанном для внешних скриптов, и запустит команду (см. параметр `ExternalScripts` в файле конфигурации сервер/прокси Zabbix). Команда будет выполнена от имени того же пользователя, что и сервер/прокси Zabbix, поэтому при необходимости все права доступа или переменные окружения следует обрабатывать в wrapper-скрипте. Права доступа к команде также должны разрешать выполнение для этого пользователя. Для выполнения доступны только команды в указанном каталоге.

Warning:
Не злоупотребляйте внешними проверками, так как для каждого скрипта серверу/прокси Zabbix требуется запуск fork-процесса, а выполнение большого количества скриптов может значительно снизить производительность Zabbix.

Пример использования

Выполнение скрипта `check_oracle.sh` с первым параметром `'-h'`. Второй параметр будет заменен IP-адресом или DNS-именем в зависимости от выбора в свойствах узла сети.

`check_oracle.sh["-h","{HOST.CONN}"]`

Предполагая, что узел сети настроен на использование IP-адреса, сервер/прокси Zabbix выполнит:

`check_oracle.sh '-h' '192.168.1.4'`

Результат внешней проверки

Возвращаемое значение внешней проверки представляет собой стандартный вывод вместе со стандартным потоком ошибок, созданным проверкой.

Attention:
Элемент данных, который возвращает текст (информация типа character, log или text), не станет неподдерживаемым в случае вывода в стандартный поток ошибок.

Возвращаемое значение ограничено 16 МБ (включая завершающие пробелы, которые обрезаются); также применяются ограничения базы данных.

Если запрошенный сценарий не найден или у сервера/прокси Zabbix нет прав на его выполнение, элемент данных станет неподдерживаемым и будет отображено соответствующее сообщение об ошибке.

В случае тайм-аута элемент данных станет неподдерживаемым, будет отображено соответствующее сообщение об ошибке, а порожденный для сценария процесс будет завершен.

Монитор базы данных

Обзор

Тип элемента данных **Монитор баз данных** в веб-интерфейсе Zabbix используется для ODBC мониторинга (проверок ODBC).

ODBC — это API на языке программирования Си к промежуточной прослойке для доступа к системам управления базами данных (DBMS). Концепт ODBC был разработан Microsoft и в дальнейшем портирован на другие платформы.

Zabbix может выполнять запросы к любой базе данных, которая поддерживается ODBC. Чтобы это сделать, Zabbix не подключается напрямую к базам данных, он использует интерфейс ODBC и драйверы, установленные в ODBC.

Это позволяет с большей эффективностью мониторить различные базы данных с разнообразными целями (например, проверка специфичных запросов к базе данных, статистика использования и прочее).

Zabbix поддерживает unixODBC, которая является одной из наиболее часто используемых реализаций API ODBC с открытым исходным кодом.

Attention:
Смотрите также известные проблемы, связанные с ODBC проверками.

Установка unixODBC

Рекомендуемый способ установки unixODBC — использовать стандартные репозитории пакетов операционной системы Linux. В большинстве популярных дистрибутивов Linux unixODBC по умолчанию включен в репозиторий пакетов. Если пакеты недоступны, исходные файлы можно получить на домашней странице unixODBC: http://www.unixodbc.org/download.html.

Чтобы установить unixODBC, используйте менеджер пакетов для выбранной вами системы:

```
#####        Ubuntu/Debian:
apt install unixodbc unixodbc-dev

#####             RedHat/Fedora:
dnf install unixODBC unixODBC-devel

#####             SUSE:
zypper in unixODBC-devel
```

Attention:
Пакет `unixodbc-dev` или `unixODBC-devel` необходим для компиляции Zabbix с поддержкой unixODBC. Чтобы включить поддержку ODBC, Zabbix следует собирать со следующей опцией конфигурации:

`--with-unixodbc[=ARG] # Use ODBC driver against unixODBC package.`

Установка драйверов unixODBC

Драйвер unixODBC базы данных должен быть установлен для базы данных, которая будет наблюдаться. Для списка поддерживаемых баз данных и драйверов обратитесь к домашней странице unixODBC: http://www.unixodbc.org/drivers.html.

Note:
В некоторых дистрибутивах Linux драйвера баз данных включены в репозитории пакетов.

MySQL

Чтобы установить драйвер unixODBC базы данных MySQL, используйте менеджер пакетов для выбранной вами системы:

```
#####        Ubuntu/Debian:
apt install odbc-mariadb

#####             RedHat/Fedora:
dnf install mariadb-connector-odbc

#####             SUSE:
zypper install mariadb-connector-odbc
```

Чтобы установить драйвер базы данных без менеджера пакетов, обратитесь к документации MySQL [en] для `mysql-connector-odbc`, или документации MariaDB [en] для `mariadb-connector-odbc`.

PostgreSQL

Чтобы установить драйвер unixODBC базы данных PostgreSQL, используйте менеджер пакетов для выбранной вами системы:

```
#####        Ubuntu/Debian:
apt install odbc-postgresql

#####             RedHat/Fedora:
dnf install postgresql-odbc
```

```
#####             SUSE:
zypper install psqlODBC
```

Чтобы установить драйвер базы данных без менеджера пакетов, обратитесь к документации PostgreSQL [en].

Oracle

Чтобы установить драйвер базы данных unixODBC, обратитесь к документации Oracle [en].

MSSQL

Чтобы установить драйвер unixODBC базы данных MSSQL, используйте менеджер пакетов для выбранной вами системы:

```
#####        Ubuntu/Debian:
apt install tdsodbc

#####             RedHat/Fedora (    EPEL: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

#####             SUSE:
zypper install libtdsodbc0
```

Чтобы установить драйвер базы данных без менеджера пакетов, обратитесь к руководству пользователя FreeTDS [en].

Настройка unixODBC

Чтобы настроить unixODBC, необходимо отредактировать файлы `odbcinst.ini` и `odbc.ini`. Вы можете проверить расположение этих файлов, выполнив следующую команду:

```
odbcinst -j
```

Результат команды должен содержать информацию наподобие следующей:

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

odbcinst.ini

Файл `odbcinst.ini` содержит список установленных драйверов баз данных ODBC. Если `odbcinst.ini` отсутствует, его необходимо создать вручную.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Параметр	Описание
TEST_MYSQL	Имя драйвера базы данных.
Description	Описание драйвера базы данных.
Driver	Местоположение библиотеки драйвера базы данных.
FileUsage	Определяет, поддерживает ли драйвер базы данных подключение к серверу базы данных без поддержки доступа к локальным файлам (0); поддерживает чтение данных из файлов (1); поддерживает запись данных в файлы (2).
Threading	Уровень сериализации потоков. Поддерживается для PostgreSQL. Начиная с версии 1.6, если менеджер драйверов собран с поддержкой потоков, вы можете добавить еще одну запись драйвера.

odbc.ini

Файл `odbc.ini` используется для настройки источников данных. Обратите внимание, что список поддерживаемых параметров зависит от драйвера базы данных (например, базы данных Oracle могут использовать ServerName вместо Server и т. д.).

```
[TEST_MYSQL]
Description=MySQL Test Database
```

```
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Параметр	Описание
TEST_MYSQL	Имя источника данных (Data source name, DSN).
Description	Описание источника данных.
Driver	Имя драйвера базы данных (как указано в `odbcinst.ini`).
Server	IP/DNS сервера базы данных.
User	Пользователь базы данных для подключения.
Password	Пароль пользователя базы данных.
Port	Порт подключения к базе данных.
Socket	Сокет подключения к базе данных.
Database	Имя базы данных.

Другие возможные параметры конфигурации смотрите в документации MySQL [en].

Файл `odbc.ini` для PostgreSQL может содержать дополнительные параметры:

```
[TEST_PSQL]
Description=PostgreSQL Test Database
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

Parameter	Description
ReadOnly	Указывает, разрешает ли подключение к базе данных только операции чтения (`SELECT`) и ограничивает изменения (`INSERT`, `UPDATE` и `DELETE`); полезно в сценариях, где данные должны оставаться неизменными.
Protocol	Версия протокола backend PostgreSQL (игнорируется при использовании SSL-соединений).
ShowOidColumn	Указывает, следует ли включать Object ID (OID) в SQLColumns.
FakeOidIndex	Указывает, следует ли создавать фиктивный уникальный индекс по OID.
RowVersioning	Указывает, следует ли включать возможность для приложений определять, были ли данные изменены другими пользователями, пока вы пытаетесь обновить строку. Обратите внимание, что этот параметр может ускорить процесс обновления, поскольку для обновления строки не требуется указывать каждый столбец в предложении `WHERE`.
ShowSystemTables	Указывает, должен ли драйвер базы данных рассматривать системные таблицы как обычные таблицы в SQLTables; полезно для доступности, обеспечивая видимость системных таблиц.
Fetch	Указывает, должен ли драйвер автоматически использовать declare cursor/fetch для обработки операторов `SELECT` и поддерживать кэш из 100 строк.
BoolsAsChar	Управляет сопоставлением логических типов. Если установлено значение "Yes", логические значения сопоставляются с `SQL_CHAR`; в противном случае — с `SQL_BIT`.

Parameter	Description
SSLmode	Указывает режим SSL для подключения.
ConnSettings	Дополнительные параметры, передаваемые backend при подключении.

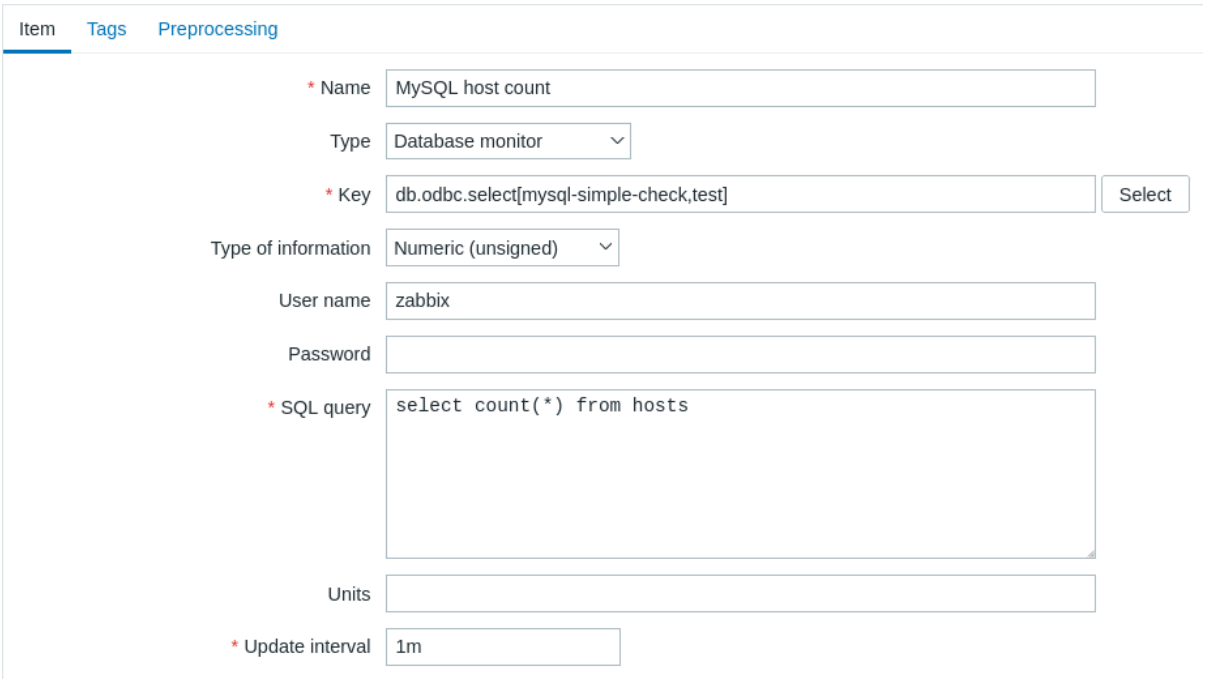
Тестирование соединения ODBC

Чтобы проверить, работает ли соединение ODBC успешно, можно использовать утилиту `isql` (входит в пакет `unixODBC`):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Настройка элемента данных в веб-интерфейсе Zabbix

Настройте элемент данных для **мониторинга базы данных**.



Все обязательные поля ввода отмечены красной звездочкой.

Для элементов данных мониторинга базы данных необходимо указать:

Type	Выберите здесь "Database monitor".
Key	Введите один из поддерживаемых ключей элемента данных: **db.odbc.select[]** - этот элемент данных возвращает одно значение (первый столбец первой строки результата SQL-запроса); **db.odbc.get[]** - этот элемент данных возвращает несколько строк/столбцов в формате JSON; **db.odbc.discovery[]** - этот элемент данных возвращает данные низкоуровневого обнаружения.
User name	Введите имя пользователя базы данных (до 255 символов). Этот параметр необязателен, если имя пользователя базы данных указано в файле `odbc.ini`. Если используется строка подключения и поле *User name* не пустое, оно добавляется к строке подключения как `UID=<user>`.

Password	Введите пароль пользователя базы данных (до 255 символов). Этот параметр необязателен, если пароль указан в файле `odbc.ini`. Если используется строка подключения и поле *Password* не пустое, оно добавляется к строке подключения как `PWD=<password>`. В этом поле поддерживаются специальные символы. Для всех баз данных, кроме Oracle, пароли, содержащие %, должны быть заключены в фигурные скобки {}. Пароль будет добавлен к строке подключения после имени пользователя, например: `UID=<username>;PWD=P?;)*word`. Чтобы проверить итоговую строку, можно выполнить следующую команду: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Введите SQL-запрос. Обратите внимание, что для `db.odbc.select[]` запрос должен возвращать только одно значение.
Type of information	Выберите здесь тип информации, который будет возвращен запросом. Если тип информации выбран неверно, элемент данных станет неподдерживаемым.

Важные примечания

- Элементы данных для мониторинга базы данных станут неподдерживаемыми, если в конфигурации сервера или прокси не запущены процессы *odbc poller*. Чтобы активировать ODBC poller, задайте параметр `StartODBCPollers` в файле конфигурации Zabbix сервера или, для проверок, выполняемых прокси, в файле конфигурации Zabbix прокси.
- Значение параметра *Timeout* в форме конфигурации элемента данных используется как тайм-аут входа в ODBC и тайм-аут выполнения запроса. Учтите, что эти параметры тайм-аута могут игнорироваться, если установленный драйвер ODBC не поддерживает их.
- Команда SQL должна возвращать набор результатов, как и любой запрос, использующий оператор `select`. Синтаксис запроса будет зависеть от СУБД, которая будет его обрабатывать. Синтаксис запроса к хранимой процедуре должен начинаться с ключевого слова `call`.

Подробности ключей элементов данных

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками **< >**, опциональны.

db.odbc.select[<уникальное короткое описание>,<dsn>,<строка подключения>]

 Возвращает одно значение, то есть первый столбец первой строки результата SQL запроса.
 Возвращаемое значение: Зависит от SQL запроса.

Параметры:

- **уникальное короткое описание** — уникальное короткое описание, чтобы идентифицировать элемент данных (для использования в триггерах и т. д.);
- **dsn** — имя источника данных (как указано в `odbc.ini`);
- **строка подключения** — строка подключения (может содержать специфичные для драйверов аргументы).

Комментарии:

- Хотя `dsn` и — опциональные параметры, по крайней мере один из них должен присутствовать; если заданы оба, то `dsn` будет игнорироваться.
- Если запрос возвращает более одного столбца, будет прочитан только первый столбец. Если запрос возвращает более одной строки, будет прочитана только первая строка.

db.odbc.get[<уникальное короткое описание>,<dsn>,<строка подключения>]

 Преобразует результат SQL-запроса в массив JSON.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **уникальное короткое описание** — уникальное короткое описание, чтобы идентифицировать элемент данных (для использования в триггерах и т. д.);
- **dsn** — имя источника данных (как указано в `odbc.ini`);
- **строка подключения** — строка подключения (может содержать специфичные для драйверов аргументы).

Комментарии:

- Хотя `dsn` и — опциональные параметры, по крайней мере один из них должен присутствовать; если заданы оба, то `dsn` будет игнорироваться.
- Может быть возвращено несколько строк/столбцов в формате JSON. Этот элемент данных можно использовать в качестве основного элемента данных, который собирает все данные за один системный вызов, а предварительную обработку JSONPath можно использовать в зависимых элементах данных для извлечения отдельных значений.

Для получения более подробных сведений смотрите пример возвращаемого формата, использованного в низкоуровневом обнаружении.

Пример:

```
#                    MySQL ODBC driver 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<уникальное короткое описание>,<dsn>,<строка подключения>]

 Преобразует результат SQL-запроса в массив JSON, используемый для низкоуровневого обнаружения. Имена столбцов из результата запроса преобразуются в имена макросов низкоуровневого обнаружения, спаренных со значениями обнаруженных полей. Эти макросы можно использовать при создании прототипов элементов данных, триггеров и т. д.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **уникальное короткое описание** — уникальное короткое описание, чтобы идентифицировать элемент данных (для использования в триггерах и т. д.);
- **dsn** — имя источника данных (как указано в `odbc.ini`);
- **строка подключения** — строка подключения (может содержать специфичные для драйверов аргументы).

Комментарии:

- Хотя `dsn` и — опциональные параметры, по крайней мере один из них должен присутствовать; если заданы оба, то `dsn` будет игнорироваться.

Сообщения об ошибках

Сообщения об ошибках ODBC структурированы по полям для предоставления подробной информации. Например, сообщение об ошибке может выглядеть следующим образом:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- «`Cannot execute ODBC query`» — сообщение Zabbix
- «`[SQL_ERROR]`» — код, возвращаемый ODBC
- «`[42601]`» — SQLState
- «`[7]`» — исходный код ошибки
- «`[ERROR: syntax error at or near ";"; Error while executing the query]`» — исходное сообщение об ошибке

Обратите внимание, что длина сообщения об ошибке ограничена 2048 байтами, поэтому сообщение может быть усечено. Если имеется более одной диагностической записи ODBC, Zabbix пытается их скомпоновать (разделяя символом `|`), насколько позволяет ограничение по длине.

HTTP агент

Обзор

Этот тип элемента данных позволяет выполнять опрос данных с использованием протокола HTTP/HTTPS. Также возможен прием данных с помощью утилиты Zabbix sender или протокола Zabbix sender (для отправки данных на сервер или прокси Zabbix), либо с помощью метода API `history.push` (для отправки данных на сервер Zabbix).

Проверки HTTP-элементов данных выполняются сервером Zabbix. Однако, если узлы сети контролируются через прокси Zabbix, проверки HTTP-элементов данных выполняются прокси.

Проверки HTTP-элементов данных не требуют запущенного агента на контролируемом узле сети.

HTTP agent поддерживает как HTTP, так и HTTPS. Zabbix при необходимости будет следовать перенаправлениям (см. параметр *Follow redirects* ниже). Максимальное число перенаправлений жестко задано и равно 10 (с использованием опции cURL `CURLOPT_MAXREDIRS`).

> **Attention:**
> Сервер/прокси Zabbix должны быть изначально настроены с поддержкой cURL (libcurl).

Проверки HTTP выполняются асинхронно — не требуется получать ответ на один запрос до начала других проверок. Разрешение DNS также выполняется асинхронно.

Максимальная параллельность асинхронных проверок составляет 1000 (определяется параметром MaxConcurrentChecksPerPoller).

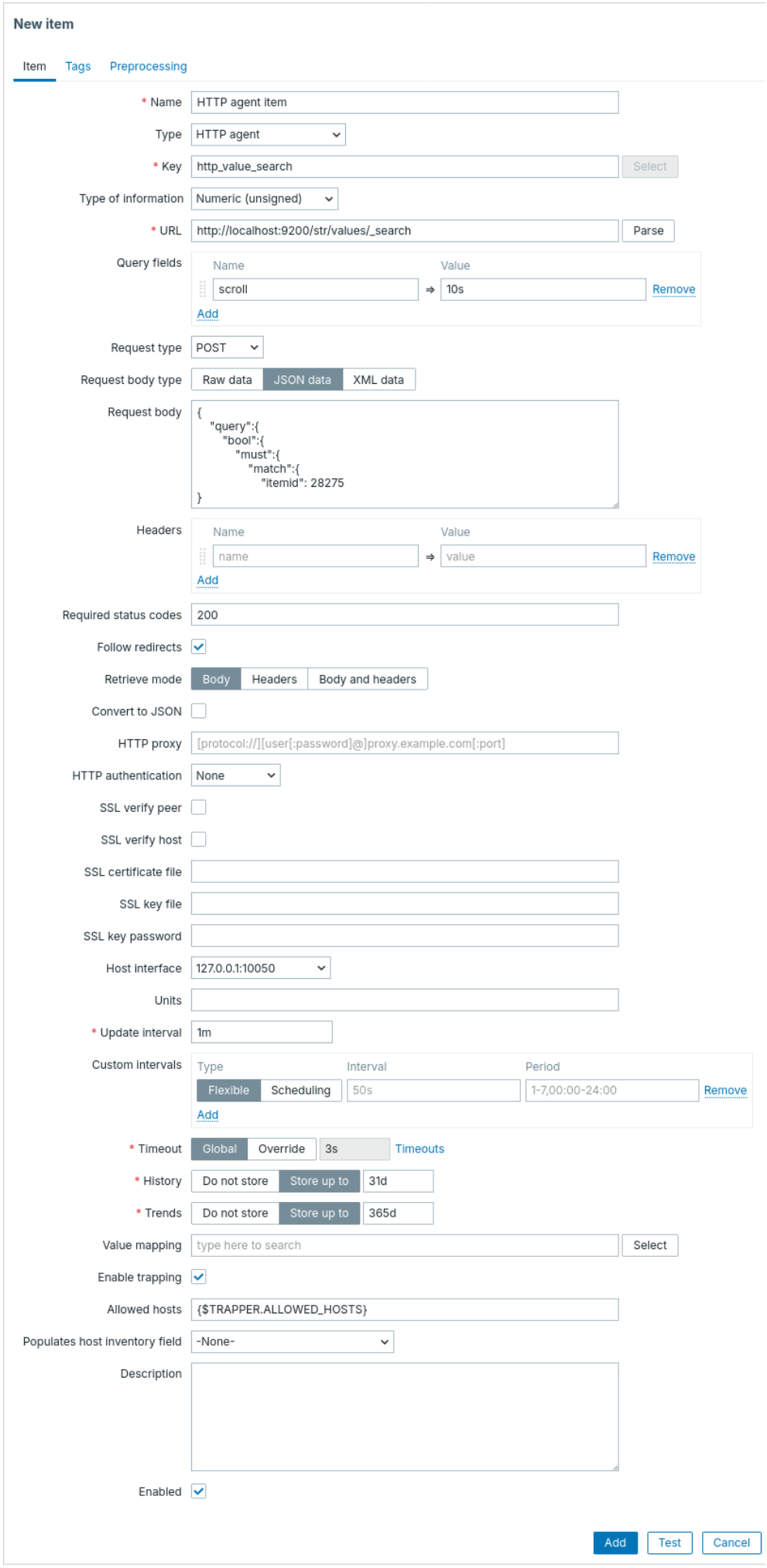
Количество асинхронных poller'ов HTTP agent определяется параметром StartHTTPAgentPollers.

Поддержка постоянных соединений cURL была добавлена в проверки HTTP agent начиная с Zabbix 7.0.

Настройка

Чтобы настроить HTTP-элемент данных:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Нажмите *Элементы данных* в строке узла сети.
3. Нажмите *Создать элемент данных*.
4. Укажите параметры элемента данных в форме.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

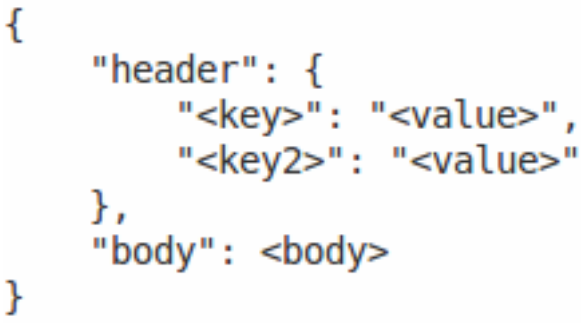
Все обязательные поля отмечены красной звездочкой.

Поля, требующие указания специальной информации для HTTP-элементов данных:

Параметр	Описание
Тип	Выберите здесь **HTTP agent**.
Ключ	Укажите уникальный ключ элемента данных.

URL	URL для подключения и получения данных. Например:
	`https://www.example.com`
	`http://www.example.com/download`
	Доменные имена можно указывать с использованием символов Unicode. При выполнении HTTP-проверки они автоматически преобразуются в ASCII с использованием punycode.
	Кнопка *Разобрать* позволяет отделить необязательные поля запроса (например, `?key=value&limit=10`) от URL, переместив атрибуты и значения в *Поля запроса* для автоматического кодирования URL.
	В целях безопасности не включайте настоящие имена пользователей или пароли в URL. Используйте пользовательские макросы, чтобы не раскрывать учетные данные.
	Ограничено 2048 символами.
	Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы обнаружения низкого уровня.
	Задается параметр cURL CURLOPT_URL.

Поля запроса	Переменные для URL (см. выше). Указываются в виде пар атрибута и значения. Значения автоматически кодируются в URL. Значения макросов сначала подставляются, а затем автоматически кодируются в URL. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL CURLOPT_URL.
Тип запроса	Выберите тип метода запроса: *GET*, *POST*, *PUT* или *HEAD*.
Тип тела запроса	Выберите тип тела запроса: **Необработанные данные** - пользовательское тело HTTP-запроса; макросы подставляются, но кодирование не выполняется. **Данные JSON** - тело HTTP-запроса в формате JSON. Макросы можно использовать как строки, числа, true и false; макросы, используемые как строки, необходимо заключать в двойные кавычки. Значения макросов сначала подставляются, а затем автоматически экранируются. Если в заголовках не указан "Content-Type", по умолчанию используется "Content-Type: application/json". **Данные XML** - тело HTTP-запроса в формате XML. Макросы можно использовать как текстовый узел, атрибут или секцию CDATA. Значения макросов сначала подставляются, а затем автоматически экранируются в текстовом узле и атрибуте. Если в заголовках не указан "Content-Type", по умолчанию используется "Content-Type: application/xml".
Тело запроса	Укажите тело запроса. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы обнаружения низкого уровня.
Заголовки	Пользовательские HTTP-заголовки, отправляемые при выполнении запроса. Указываются в виде пар атрибута и значения. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL CURLOPT_HTTPHEADER.
Требуемые коды состояния	Список ожидаемых кодов состояния HTTP. Если сервер Zabbix получает код, отсутствующий в списке, элемент данных становится неподдерживаемым. Если список пуст, проверка не выполняется. Например: 200,201,210-299. Поддерживаемые макросы в списке: пользовательские макросы, макросы обнаружения низкого уровня. Используется параметр cURL CURLINFO_RESPONSE_CODE.
Следовать перенаправлениям	Установите флажок, чтобы следовать HTTP-перенаправлениям. Задается параметр cURL CURLOPT_FOLLOWLOCATION.

Параметр	Описание
Режим получения	Выберите часть ответа, которую необходимо получить: **Только тело** - только тело **Только заголовки** - только заголовки **Тело и заголовки** - тело и заголовки
Преобразовать в JSON	Заголовки сохраняются как пары атрибута и значения под ключом "header", если для *Режим получения* выбрано значение, отличное от *Только тело*. Если обнаружено 'Content-Type: application/json', тело сохраняется как объект; в противном случае оно сохраняется как строка, например: 
HTTP-прокси	Можно указать используемый HTTP-прокси в формате `[protocol://][username[:password]@]proxy.example.com[:port]`. Необязательный префикс `protocol://` можно использовать для указания альтернативных протоколов прокси (например, `https`, `socks4`, `socks5`; см. документацию; поддержка префикса протокола добавлена в cURL 7.21.7). Если протокол не указан, прокси будет считаться HTTP-прокси. Если указать неправильный протокол, соединение завершится ошибкой, а элемент данных станет неподдерживаемым. По умолчанию используется порт 1080. Если прокси указан, он переопределяет переменные окружения, связанные с прокси, например `http_proxy`, `HTTPS_PROXY`. Если прокси не указан, переменные окружения, связанные с прокси, не переопределяются. Введенное значение передается «как есть», проверка корректности не выполняется. Обратите внимание, что для HTTP-прокси поддерживается только простая аутентификация. Поддерживаемые макросы: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL `CURLOPT_PROXY`.
HTTP-аутентификация	Выберите вариант аутентификации: **Нет** - аутентификация не используется; **Basic** - используется базовая аутентификация; **NTLM** - используется аутентификация NTLM (Windows NT LAN Manager); **Kerberos** - используется аутентификация Kerberos (см. также: Настройка Kerberos с Zabbix); **Digest** - используется дайджест-аутентификация. Задается параметр cURL `CURLOPT_HTTPAUTH`.
Имя пользователя	Укажите имя пользователя (до 255 символов). Это поле доступно, если для *HTTP-аутентификация* выбрано значение *Basic*, *NTLM*, *Kerberos* или *Digest*. Поддерживаются пользовательские макросы и макросы обнаружения низкого уровня.
Пароль	Укажите пароль пользователя (до 255 символов). Это поле доступно, если для *HTTP-аутентификация* выбрано значение *Basic*, *NTLM*, *Kerberos* или *Digest*. Поддерживаются пользовательские макросы и макросы обнаружения низкого уровня.
Проверять SSL-узел	Установите флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера проверяется по сертификатам доверенных центров сертификации (CA) из общесистемного расположения CA. Расположение файлов CA можно переопределить с помощью параметра конфигурации сервера Zabbix или прокси SSLCALocation. Задается параметр cURL `CURLOPT_SSL_VERIFYPEER`.
Проверять SSL-узел	Установите флажок, чтобы проверять соответствие поля *Common Name* или *Subject Alternate Name* сертификата веб-сервера. Задается параметр cURL `CURLOPT_SSL_VERIFYHOST`.

Параметр	Описание
Файл SSL-сертификата	Имя файла SSL-сертификата, используемого для аутентификации клиента. Файл сертификата должен быть в формате PEM[1]. Если файл сертификата также содержит закрытый ключ, оставьте поле *Файл SSL-ключа* пустым. Если ключ зашифрован, укажите пароль в поле *Пароль SSL-ключа*. Каталог, содержащий этот файл, задается параметром конфигурации сервера Zabbix или прокси `SSLCertLocation`. Поддерживаемые макросы: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL `CURLOPT_SSLCERT`.
Файл SSL-ключа	Имя файла закрытого SSL-ключа, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM[1]. Каталог, содержащий этот файл, задается параметром конфигурации сервера Zabbix или прокси `SSLKeyLocation`. Поддерживаемые макросы: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL `CURLOPT_SSLKEY`.
Пароль SSL-ключа	Пароль файла закрытого SSL-ключа. Поддерживаемые макросы: пользовательские макросы, макросы обнаружения низкого уровня. Задается параметр cURL `CURLOPT_KEYPASSWD`.
Тайм-аут	Сервер Zabbix не будет тратить на обработку URL больше указанного времени (1-600 секунд). Фактически этот параметр определяет максимальное время установления соединения с URL и максимальное время выполнения HTTP-запроса. Поэтому сервер Zabbix не будет тратить на одну проверку более 2 х *Тайм-аут* секунд. Задается параметр cURL `CURLOPT_TIMEOUT`. Дополнительную информацию о параметре *Тайм-аут* см. в разделе общие атрибуты элементов данных.
Включить прием данных	Если этот флажок установлен, элемент данных также будет работать как элемент данных-траппер и принимать данные, отправленные на сервер Zabbix или прокси с помощью утилиты Zabbix sender или протокола Zabbix sender, либо принимать данные, отправленные на сервер Zabbix с помощью метода API `history.push`. Дополнительную информацию об отправке данных см. в разделе: Элементы данных-трапперы.
Разрешенные узлы сети	Отображается только если установлен флажок *Включить прием данных*. Список IP-адресов, разделенных запятыми (необязательно в нотации CIDR), или DNS-имен. Значение по умолчанию - пользовательский макрос `{$TRAPPER.ALLOWED_HOSTS}`. По умолчанию `{$TRAPPER.ALLOWED_HOSTS}` имеет значение `127.0.0.1,::1`. Можно указать адреса непосредственно в поле или изменить значение макроса. Если оставить поле пустым, входящие соединения будут запрещены для всех узлов сети. Если значение указано, входящие соединения будут приниматься только от перечисленных здесь узлов сети. Если поддержка IPv6 включена, `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` считаются одинаковыми, а `::/0` разрешает любой IPv4- или IPv6-адрес. `0.0.0.0/0` можно использовать для разрешения любого IPv4-адреса. Обратите внимание, что «IPv4-совместимые IPv6-адреса» (префикс `0000::/96`) поддерживаются, но объявлены устаревшими в RFC4291. Пример: `127.0.0.1`, `192.168.1.0/24`, `192.168.3.1-255`, `192.168.1-10.1-255`, `::1,2001:db8::/32`, `mysqlserver1`, `zabbix.example.com`, `{HOST.HOST}` В этом поле разрешены пробелы и пользовательские макросы. В этом поле разрешены макросы узла сети: `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.IP}`, `{HOST.DNS}`, `{HOST.CONN}`.

Note:
Если поле *HTTP-прокси* оставить пустым, для использования HTTP-прокси можно задать переменные окружения, связанные с прокси.
Для HTTP - задайте переменную окружения `http_proxy` для пользователя сервера Zabbix. Например:
 `http_proxy=http://proxy_ip:proxy_port`.
Для HTTPS - задайте переменную окружения `HTTPS_PROXY`. Например:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Дополнительные сведения можно получить, выполнив команду оболочки: *# man curl*.

Attention:
[1] Zabbix поддерживает файлы сертификатов и закрытых ключей только в формате PEM. Если данные сертификата и закрытого ключа находятся в файле формата PKCS #12 (обычно с расширением `*.p12` или `*.pfx`), можно создать из него файл PEM с помощью следующих команд:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Примеры

Пример 1

Отправляйте простые GET-запросы для получения данных от таких сервисов, как Elasticsearch:

- Создайте элемент данных GET с URL: `localhost:9200/?pretty`
- Обратите внимание на ответ:

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- Теперь извлеките номер версии с помощью шага предварительной обработки JSONPath: `$.version.number`

Пример 2

Отправляйте простые POST-запросы для получения данных из таких сервисов, как Elasticsearch:

- Создайте элемент данных POST с URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Настройте следующий POST body, чтобы получить загрузку процессора (среднее за 1 минуту на ядро)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Получено:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
```

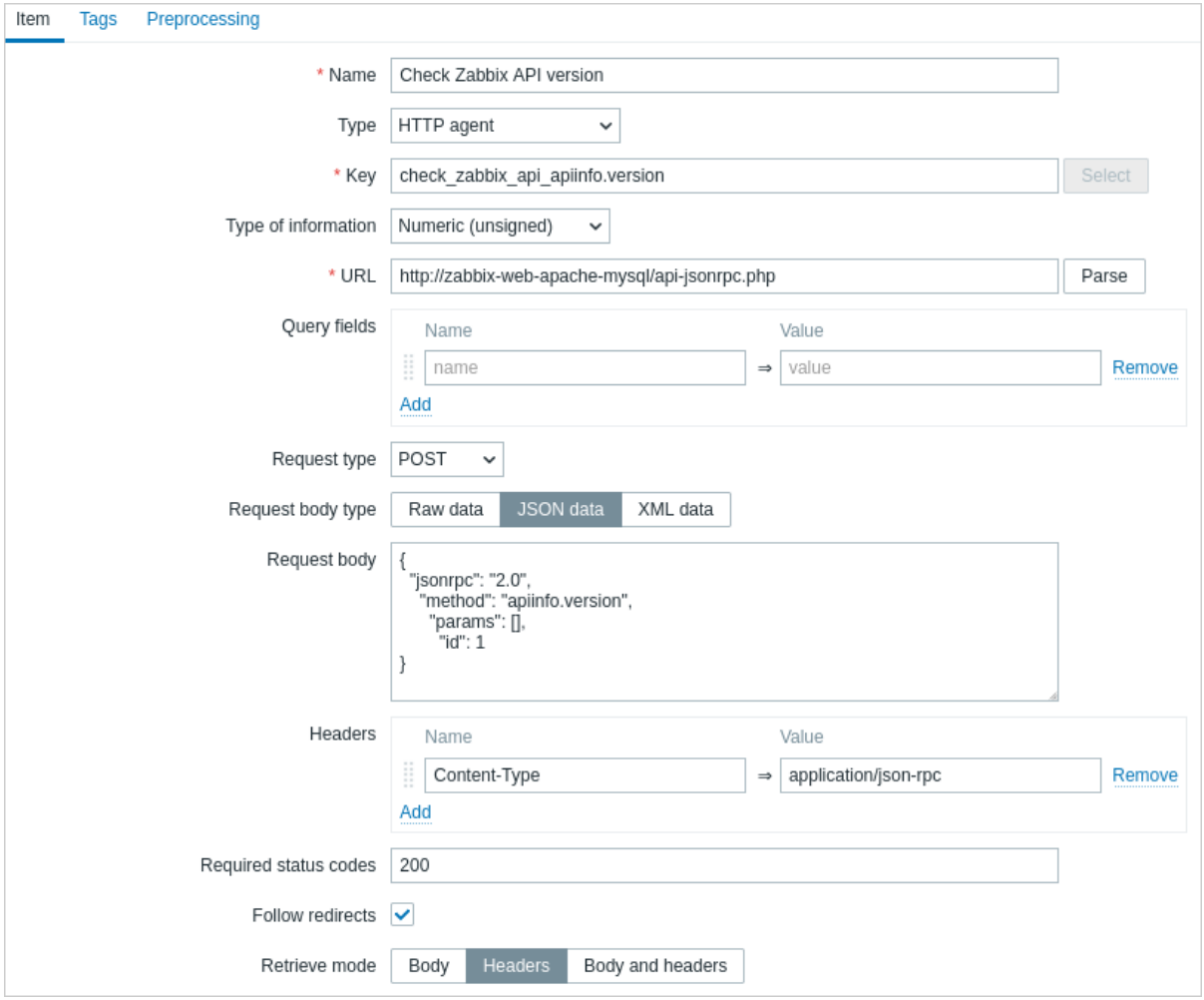
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```


- Теперь используйте шаг предварительной обработки JSONPath, чтобы получить значение элемента данных: `$.hits.hits[0]._source.value`

Пример 3

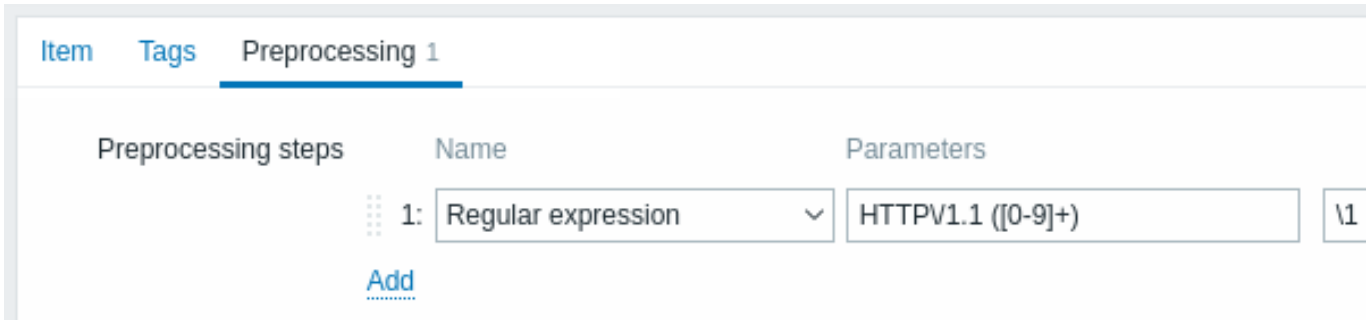
Проверка доступности API Zabbix с использованием apiinfo.version.

- Конфигурация элемента данных:

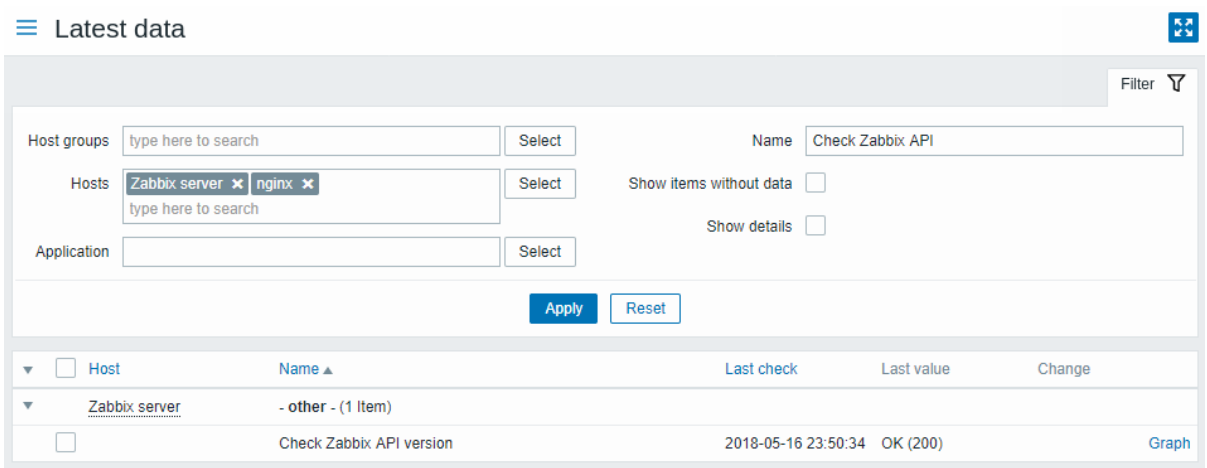


Обратите внимание на использование метода POST с JSON-данными, настройку заголовков запроса и указание возвращать только заголовки:

- Предобработка значения элемента данных с помощью регулярного выражения для получения кода HTTP:



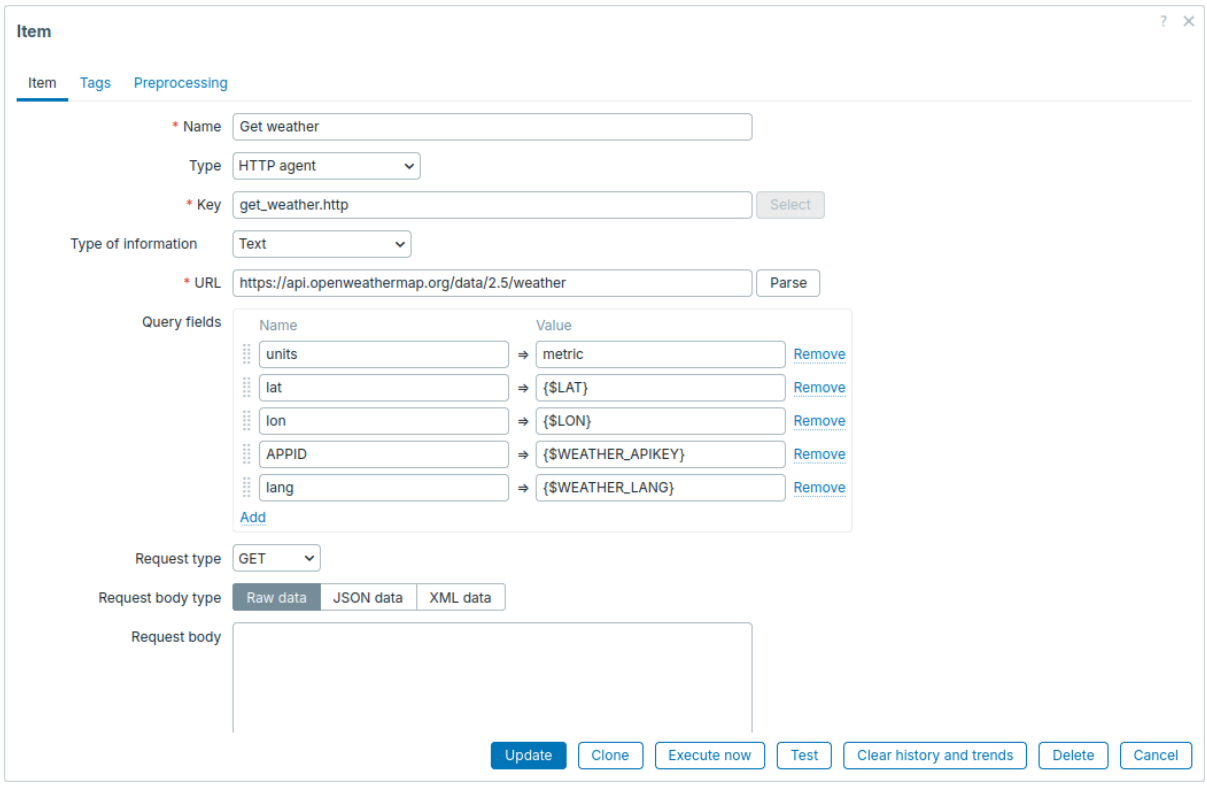
- Проверка результата в *Последние данные*:



Пример 4

Получение информации о погоде путем подключения к общедоступному сервису Openweathermap.

- Настройте мастер-элемент для массового сбора данных в одном JSON:



Обратите внимание на использование макросов в полях запроса. См. Openweathermap API для информации о том, как их заполнить.

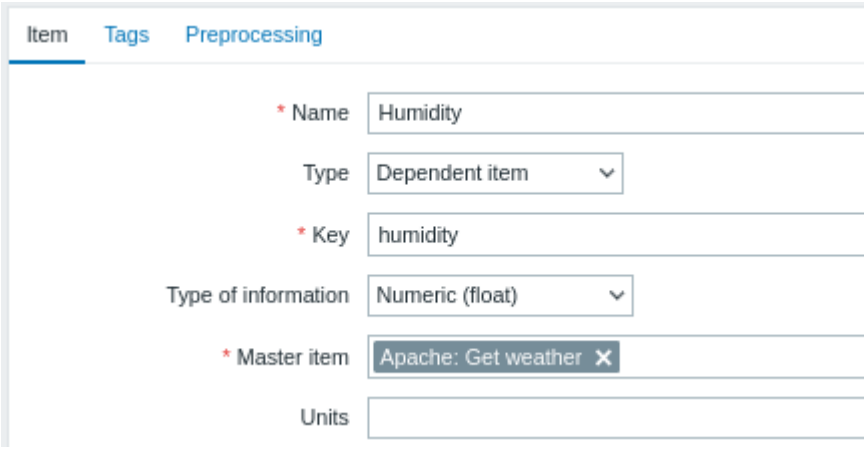
Пример JSON, возвращаемый в ответ HTTP-агентом:

```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
```

```
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
    }
}
```

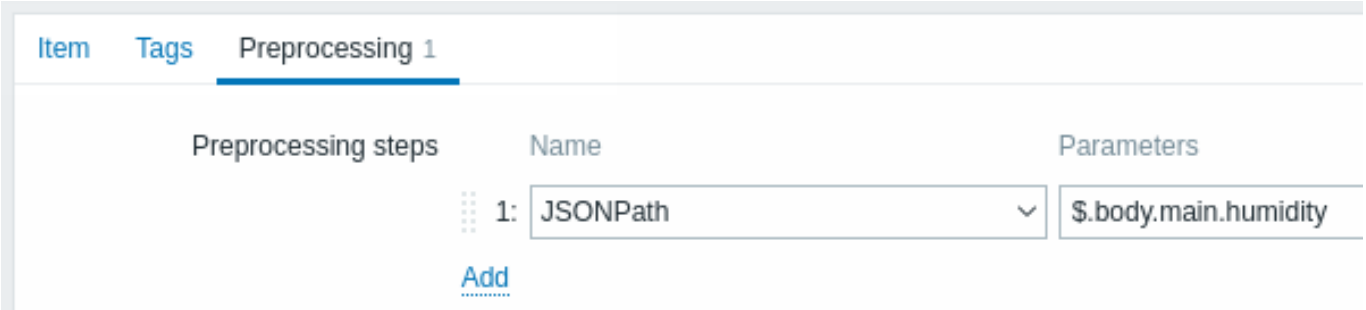

Следующая задача — настроить зависимые элементы, которые извлекают данные из JSON.

- Настройте пример зависимого элемента для влажности:

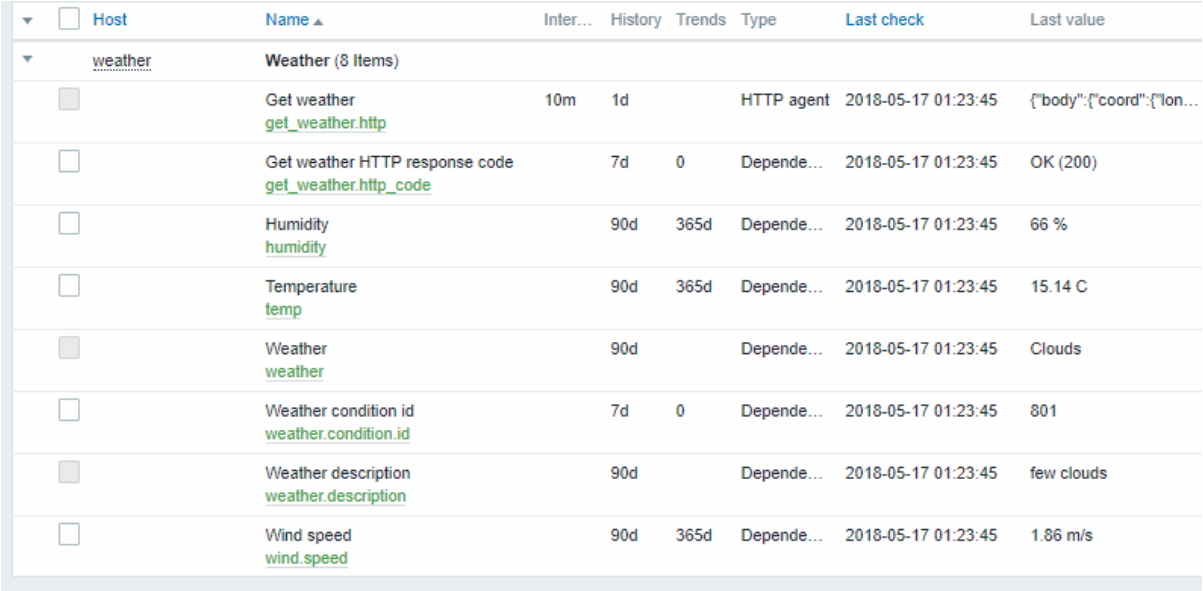


Другие погодные показатели, такие как 'Temperature', добавляются аналогичным образом.

- Пример предварительной обработки значения зависимого элемента с использованием JSONPath:



- Проверьте результат данных о погоде в *Latest data*:

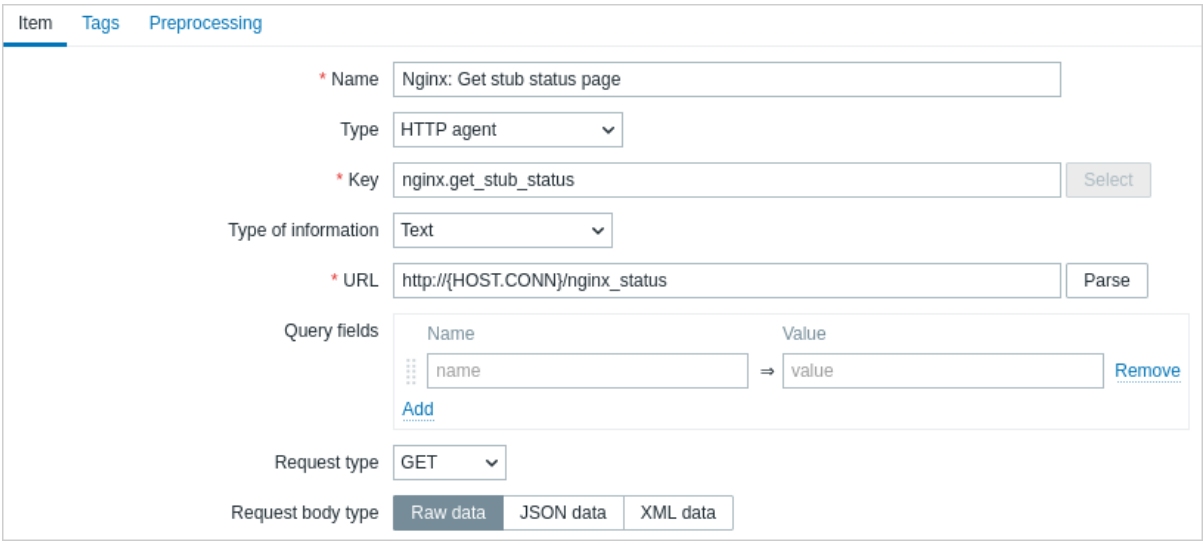


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

Пример 5

Подключение к странице состояния NGINX и получение его метрик пакетно.

- Настройте NGINX в соответствии с официальным руководством.
- Настройте главный элемент данных для пакетного сбора данных:

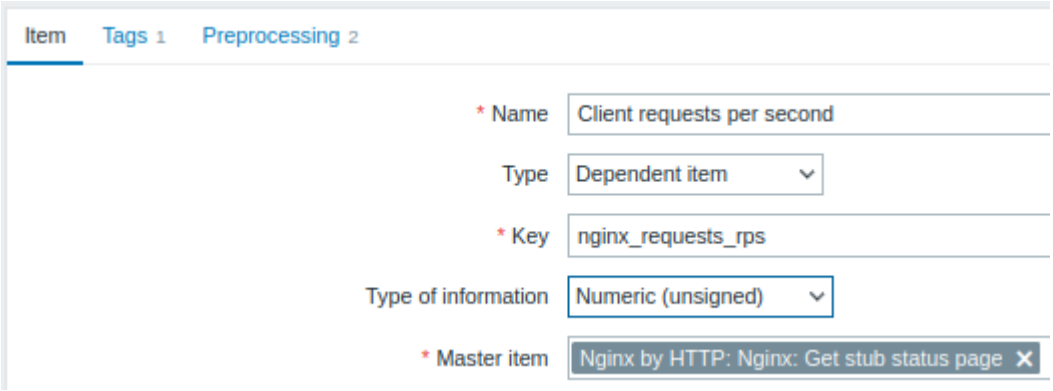


Пример вывода статуса NGINX:

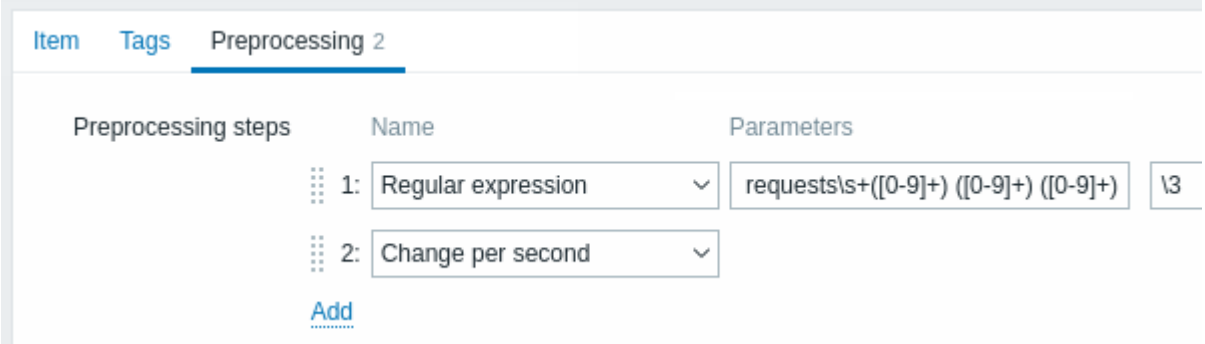
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Следующая задача — настроить зависимые элементы данных, извлекающие данные.

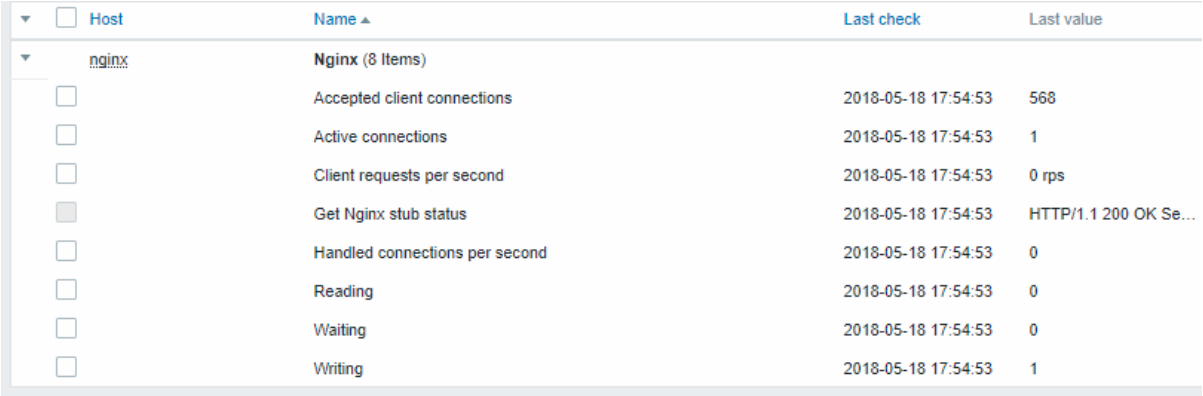
- Настройте пример зависимого элемента данных для запросов в секунду:



- Пример предварительной обработки значения зависимого элемента данных с помощью регулярного выражения `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Проверьте полный результат работы модуля состояния в разделе *Последние данные*:



Проверка Prometheus

Обзор

Zabbix может запрашивать метрики, предоставляемые в формате строк Prometheus.

Чтобы начать сбор данных Prometheus, необходимо выполнить два шага:

- HTTP master item, указывающий на соответствующую конечную точку данных, например `https://<prometheus host>/metrics`
- зависимые элементы данных, использующие опцию предварительной обработки Prometheus для запроса необходимых данных из метрик, собранных master item

Существует две опции предварительной обработки данных Prometheus:

- *Prometheus pattern* - используется в обычных элементах данных для запроса данных Prometheus
- *Prometheus to JSON* - используется в обычных элементах данных и для низкоуровневого обнаружения. В этом случае запрошенные данные Prometheus возвращаются в формате JSON.

Пакетная обработка

Пакетная обработка поддерживается для зависимых элементов данных. Чтобы включить кэширование и индексирование, предварительная обработка *Prometheus pattern* должна быть **первым** шагом предварительной обработки. Когда

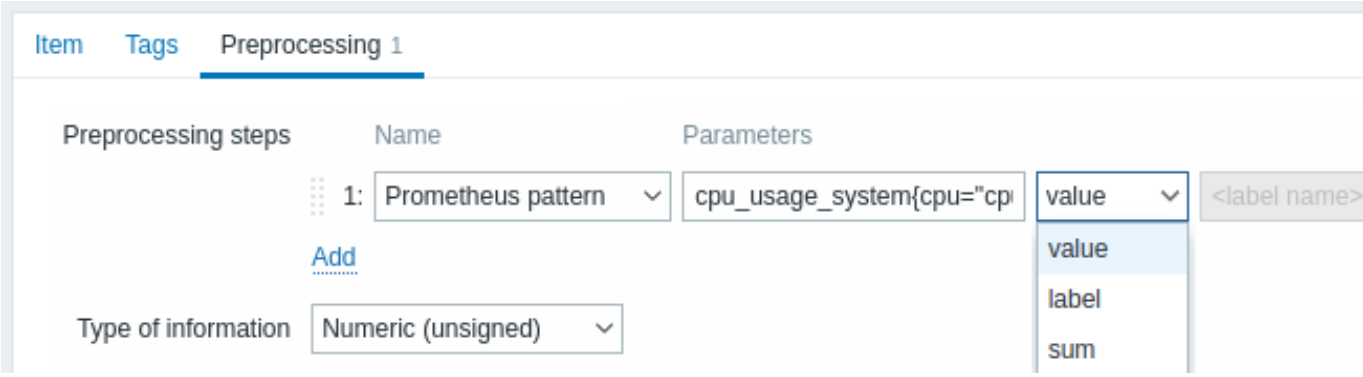
Prometheus pattern является первым шагом предварительной обработки, то разобранные данные Prometheus кэшируются и индексируются по первому условию <label>==<value> в шаге предварительной обработки *Prometheus pattern*. Этот кэш повторно используется при обработке других зависимых элементов данных в этом пакете. Для оптимальной производительности первым должен быть тот label, у которого больше всего различных значений.

Если перед первым шагом требуется выполнить другую предварительную обработку, ее следует перенести либо в основной элемент данных, либо в новый зависимый элемент данных, который будет использоваться как основной элемент данных для зависимых элементов данных.

Настройка

При условии, что основной элемент данных HTTP уже настроен, необходимо создать зависимый элемент данных, использующий шаг предварительной обработки Prometheus:

1. Введите общие параметры зависимого элемента данных в форме настройки.
2. Перейдите на вкладку *Предварительная обработка*.
3. Выберите вариант предварительной обработки Prometheus (*Шаблон Prometheus* или *Prometheus в JSON*).



Следующие параметры относятся к варианту предварительной обработки *Шаблон Prometheus*:

Параметр	Описание	Примеры
Шаблон	Для определения требуемого шаблона данных можно использовать язык запросов, похожий на язык запросов Prometheus (см. таблицу сравнения), например: <metric name> - выбор по имени метрики {__name__="<metric name>"} - выбор по имени метрики {__name__=~"<regex>"} - выбор по имени метрики, соответствующему регулярному выражению {<label name>="<label value>",...} - выбор по имени метки {<label name>=~"<regex>",...} - выбор по имени метки, соответствующему регулярному выражению {__name__=~".*"}==<value> - выбор по значению метрики Или комбинацию указанных вариантов: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Значением метки может быть любая последовательность символов UTF-8, однако символы обратной косой черты, двойной кавычки и перевода строки необходимо экранировать как \\, \" и \n соответственно; остальные символы экранировать не следует.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=".*"}* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

Параметр	Описание	Примеры
Обработка результата	Укажите, что необходимо вернуть: значение, метку или результат соответствующей функции (если шаблон соответствует нескольким строкам и результат необходимо агрегировать): **value** - вернуть значение метрики (ошибка, если найдено несколько строк) **label** - вернуть значение метки, указанной в поле *Метка* (ошибка, если найдено несколько метрик) **sum** - вернуть сумму значений **min** - вернуть минимальное значение **max** - вернуть максимальное значение **avg** - вернуть среднее значение **count** - вернуть количество значений Это поле доступно только для варианта *Шаблон Prometheus*.	См. также примеры использования параметров ниже.
Вывод	Укажите имя метки (необязательно). В этом случае возвращается значение, соответствующее имени метки. Это поле доступно только для варианта *Шаблон Prometheus*, если в поле *Обработка результата* выбран вариант «Метка».	

Примеры использования параметров

1. Наиболее распространенный вариант использования - возврат **значения**. Чтобы вернуть значение /var/db из:

   node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09
 node_disk_usage_bytes{path="/var/db"} 20480
 node_disk_usage_bytes{path="/var/dpkg"} 8192
 node_disk_usage_bytes{path="/var/empty"} 4096

используйте следующие параметры:

- *Шаблон* - node_disk_usage_bytes{path="/var/db"}
- *Обработка результата* - выберите «value»

2. Также может потребоваться **среднее** значение всех параметров node_disk_usage_bytes:

- *Шаблон* - node_disk_usage_bytes
- *Обработка результата* - выберите «avg»

3. Хотя Prometheus поддерживает только числовые данные, для возврата соответствующего текстового описания также часто используется обходное решение. Для этого можно применить фильтр и указать метку. Чтобы вернуть значение метки «color» из:

   elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1
 elasticsearch_cluster
   0
 elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0

используйте следующие параметры:

- *Шаблон* - elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1
- *Обработка результата* - выберите «label»
- *Метка* - укажите «color»

Фильтр (основанный на числовом значении «1») выберет соответствующую строку, а метка вернет описание состояния (в настоящее время «green», но потенциально также «red» или «yellow»).

Prometheus в JSON

Данные из Prometheus можно использовать для низкоуровневого обнаружения. В этом случае требуются данные в формате JSON, и опция предварительной обработки *Prometheus to JSON* вернет именно их.

Подробнее см. Обнаружение с использованием данных Prometheus.

Сравнение языков запросов

В следующей таблице перечислены различия и сходства между PromQL и языком запросов предварительной обработки Zabbix для Prometheus.

Селектор мгновенного вектора PromQL		Предварительная обработка Zabbix для Prometheus
Различия		
Целевой объект запроса	сервер Prometheus	Обычный текст в формате экспозиции Prometheus
Возвращает	Мгновенный вектор	Значение метрики или метки (шаблон Prometheus) Массив метрик для одного значения в JSON (Prometheus to JSON)
Операторы сопоставления меток	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Регулярное выражение, используемое при сопоставлении имени метки или метрики	RE2	PCRE
Операторы сравнения	См. список	Для фильтрации значений поддерживается только **==** (равно)
Сходства		
Выбор по имени метрики, которое равно строке	<metric name> or {__name__="<metric name>"}	<metric name> or {__name__="<metric name>"}
Выбор по имени метрики, которое соответствует регулярному выражению	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Выбор по значению <label name>, которое равно строке	{<label name>="<label value>",...}	{<label name>="<label value>",...}
Выбор по значению <label name>, которое соответствует регулярному выражению	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
Выбор по значению, которое равно строке	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

IPMI агент

Обзор

В Zabbix можно отслеживать состояние и доступность устройств с интерфейсом Intelligent Platform Management Interface (IPMI). Для выполнения проверок IPMI сервер Zabbix необходимо предварительно настроить с поддержкой IPMI.

IPMI — это стандартизированный интерфейс для удаленного управления компьютерными системами в режиме «lights-out» или «внеполосного» управления. Он позволяет отслеживать состояние оборудования непосредственно с так называемых карт «внеполосного» управления, независимо от операционной системы и от того, включен ли компьютер.

Мониторинг IPMI в Zabbix работает только для устройств с поддержкой IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP и т. д.).

Процесс управления IPMI планирует проверки IPMI, выполняемые опросчиками IPMI. В каждый момент времени узел сети опрашивается только одним опросчиком IPMI, что уменьшает количество открытых подключений к контроллерам BMC. Поэтому количество опросчиков IPMI можно безопасно увеличить, не опасаясь перегрузки контроллеров BMC. Процесс управления IPMI автоматически запускается при запуске хотя бы одного опросчика IPMI.

См. также известные проблемы, связанные с проверками IPMI.

Настройка

Настройка узла сети

Узел сети необходимо настроить для обработки проверок IPMI. Нужно добавить интерфейс IPMI, с соответствующими IP адресом и номером порта, а также задать параметры аутентификации IPMI.

Смотрите настройку узлов сети для получения более подробных сведений.

Конфигурация сервера

По умолчанию сервер Zabbix не настроен на запуск каких-либо опрашивателей IPMI, поэтому любые добавленные элементы данных IPMI не будут работать. Чтобы изменить это, откройте файл конфигурации сервера Zabbix (`zabbix_server.conf`) от имени root и найдите следующую строку:

```
# StartIPMIPollers=0
```

Удалите символ комментария и установите количество опрашивателей, например, в 3, чтобы строка выглядела так:

```
StartIPMIPollers=3
```

Сохраните файл и затем перезапустите zabbix_server.

Настройка элемента данных

Для настройки элемента данных на уровне узла сети:

- В поле *Тип* выберите «IPMI агент»
- Введите ключ элемента данных, уникальный в пределах узла сети (скажем, ipmi.fan.rpm)
- В поле *Интерфейс узла сети* выберите подходящий IPMI интерфейс (IP и порт). Обратите внимание, IPMI интерфейс должен уже существовать на узле сети.
- Укажите *IPMI датчик*, с которого забирать метрику (например, «FAN MOD 1A RPM» на Dell Poweredge). По умолчанию, необходимо указать ID датчика. Также имеется возможность использования префиксов до самого значения:
  - `id:` — чтобы указать ID датчика;
  - `name:` — чтобы указать полное имя датчика. Эта опция может быть полезна в ситуациях, когда датчики можно отличить, только указав полное имя.
- Выберите соответствующий тип информации («Числовой (с плавающей точкой)» в данном случае; для дискретных датчиков — «Числовой (целое положительное)»), единицы измерения (скорее всего, «rpm») и любые другие требуемые атрибуты элемента данных.

Поддерживаемые проверки

IPMI агент поддерживает встроенный элемент данных **ipmi.get**, который возвращает информацию, связанную с датчиком IPMI, и может использоваться для обнаружения датчиков IPMI.
 Возвращаемое значение: *объект JSON*

Время ожидания и завершение сессии

Время ожидания IPMI сообщений и количества попыток определены в библиотеке OpenIPMI. В связи с текущим дизайном OpenIPMI, невозможно сделать эти значения настраиваемыми из Zabbix, ни на уровне интерфейса, ни на уровне элемента данных.

Время ожидания неактивности IPMI сессии для LAN равняется 60 +/-3 секунд. В настоящее время невозможно реализовать периодическую отправку команды Активации Сессии в OpenIPMI. Если проверки IPMI элементов данных от Zabbix к конкретному BMC не выполняются в течение времени, большего, чем время ожидания сессии, настроенное в BMC, то следующая проверка IPMI после истечения времени ожидания приведёт к ошибкам из-за превышения времени ожидания отдельного сообщения, повторных попыток или ошибке при получении. После этого открывается новая сессия и инициируется полное повторное сканирование BMC. Для управления новой сессией может быть открыт новый порт UDP. Неактивность определяется как отсутствие и исходящих запросов, и входящих ответов. Если вы хотите избежать лишнего сканирования BMC, рекомендуется установить интервал опроса элементов данных IPMI ниже времени ожидания неактивности IPMI сессии, настроенного в BMC.

Примечания по дискретным датчикам IPMI

Чтобы найти датчики на узле сети, запустите сервер Zabbix с включенным **DebugLevel=4**. Подождите несколько минут и найдите записи обнаружения датчиков в файле журнала сервера Zabbix:

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

Чтобы расшифровать типы и состояния датчиков IPMI, доступна копия спецификации IPMI 2.0 (обратите внимание, что дальнейшие обновления спецификации IPMI не планируются).

Первый параметр, с которого следует начать, — "reading_type". Используйте "Table 42-1, Event/Reading Type Code Ranges" из спецификации, чтобы расшифровать код "reading_type". Большинство датчиков в нашем примере имеют "reading_type:0x1", что означает датчик типа "threshold". "Table 42-3, Sensor Type Codes" показывает, что "type:0x1" означает датчик температуры, "type:0x2" - датчик напряжения, "type:0x4" - вентилятор и т. д. Датчики threshold иногда называют "analog" датчиками, поскольку они измеряют непрерывные параметры, такие как температура, напряжение, обороты в минуту.

Другой пример — датчик с "reading_type:0x3". "Table 42-1, Event/Reading Type Code Ranges" говорит, что коды типа чтения 02h-0Ch означают датчик "Generic Discrete". Дискретные датчики имеют до 15 возможных состояний (другими словами — до 15 значимых битов). Например, для датчика 'CATERR' с "type:0x7" "Table 42-3, Sensor Type Codes" показывает, что этот тип означает "Processor", а значение отдельных битов такое: 00h (младший значащий бит) — IERR, 01h — Thermal Trip и т. д.

В нашем примере есть несколько датчиков с "reading_type:0x6f". Для этих датчиков "Table 42-1, Event/Reading Type Code Ranges" рекомендует использовать "Table 42-3, Sensor Type Codes" для расшифровки значений битов. Например, датчик 'Power Unit Stat' имеет тип "type:0x9", что означает "Power Unit". Offset 00h означает "PowerOff/Power Down". Иными словами, если младший значащий бит равен 1, то сервер выключен. Чтобы проверить этот бит, можно использовать функцию `bitand` с маской '1'. Выражение триггера может выглядеть так:

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

чтобы предупредить о выключении сервера.

Заметки об именах дискретных датчиков в OpenIPMI-2.0.16, 2.0.17, 2.0.18 и 2.0.19

Имена дискретных датчиков в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 зачастую имеют дополнительный символ «0» (или какую-то другую цифру или символ), добавленный в конце имени. Например, тогда как `ipmitool` и OpenIPMI-2.0.19 отображают имена датчиков как «`PhysicalSecurity`» или «`CATERR`», в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 эти имена — «`PhysicalSecurity0`» или «`CATERR0`» соответственно.

При настройке элемента данных IPMI для Zabbix сервера, использующего OpenIPMI-2.0.16, 2.0.17 и 2.0.18, добавьте к их именам «0» в поле *IPMI датчик* для элементов данных IPMI агента. Когда ваш Zabbix сервер будет обновлён в новом Linux дистрибутиве, использующем OpenIPMI-2.0.19 (или более позднюю), элементы данных с такими IPMI дискретными датчиками перейдут в состояние «НЕ ПОДДЕРЖИВАЕТСЯ». Вам потребуется изменить их имена *IPMI датчик* (удалить «0» в конце) и подождать некоторое время, пока они станут «Активированными» снова.

Заметки об одновременной доступности пороговых и дискретных датчиков

Некоторые IPMI агенты предоставляют одновременно пороговые и дискретные датчики под одним именем. Предпочтение всегда отдаётся пороговому датчику.

Примечания о завершении соединений

Если IPMI проверки не выполняются (по любой из причин: все элементы данных IPMI деактивированы / не поддерживаются на узле сети, сам узел сети деактивирован/удалён, узел сети находится в режиме обслуживания и так далее), то соединение будет разорвано со стороны Zabbix сервера и прокси через 3 — 4 часа, в зависимости от времени запуска Zabbix сервера/прокси.

SSH агент

Обзор

Проверки SSH выполняются как мониторинг без агента. Для проверок SSH агент Zabbix не требуется.

Для выполнения проверок SSH сервер Zabbix должен быть изначально настроен с поддержкой SSH2 (libssh или libssh2). См. также: Требования.

> **Attention:**
> Начиная с RHEL 8 поддерживается только libssh. Для других дистрибутивов рекомендуется использовать libssh вместо libssh2.

Настройка

Аутентификация по паролю

Проверки SSH предоставляют два метода аутентификации: пара логин пользователя/пароль и на основе файла ключа.

Если вы не собираетесь использовать ключи, то (помимо привязки библиотеки libssh2/libssh к Zabbix при компиляции из исходных кодов) никакой дополнительной настройки не требуется.

Аутентификация с помощью файла ключа

Чтобы использовать аутентификацию по ключу для элементов данных SSH, требуются определенные изменения в конфигурации сервера.

Откройте файл конфигурации сервера Zabbix (`zabbix_server.conf`) от имени `root` и найдите следующую строку:

```
##### SSHKeyLocation=
```

Раскомментируйте ее и укажите полный путь к папке, где будут находиться открытый и закрытый ключи:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Сохраните файл и затем перезапустите сервер Zabbix.

Путь `/home/zabbix` здесь — это домашний каталог учетной записи *zabbix*, а `.ssh` — это каталог, в котором по умолчанию будут сгенерированы открытый и закрытый ключи с помощью команды ssh-keygen внутри домашнего каталога.

Обычно пакеты установки сервера Zabbix из разных дистрибутивов ОС создают учетную запись *zabbix* с домашним каталогом в другом месте, например, `/var/lib/zabbix` (как для системных учетных записей).

Перед генерацией ключей вы можете переназначить домашний каталог на `/home/zabbix`, чтобы он соответствовал параметру конфигурации `SSHKeyLocation` сервера Zabbix, упомянутому выше.

> **Note:**
> Следующие шаги можно пропустить, если учетная запись *zabbix* была добавлена вручную в соответствии с разделом установки. В таком случае домашний каталог учетной записи *zabbix* скорее всего уже `/home/zabbix`.

Чтобы изменить домашний каталог учетной записи *zabbix*, необходимо остановить все рабочие процессы, которые его используют:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Чтобы изменить расположение домашнего каталога с попыткой перенести его (если он существует), следует выполнить следующую команду:

```
usermod -m -d /home/zabbix zabbix
```

Также возможно, что домашний каталог не существовал в старом расположении, поэтому его следует создать в новом месте. Безопасная попытка сделать это:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Чтобы убедиться, что все защищено должным образом, можно выполнить дополнительные команды для установки прав доступа к домашнему каталогу:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Теперь ранее остановленные процессы можно снова запустить:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Теперь можно выполнить шаги по генерации открытого и закрытого ключей с помощью следующих команд (для удобства чтения приглашения командной строки опущены):

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
```

```
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |        .        |
##### |       o         |
##### | .      o        |
##### |+      . S       |
##### |.+    o =        |
##### |E .    * =       |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
Открытый и закрытый ключи (id_rsa.pub и id_rsa) по умолчанию были сгенерированы в каталоге /home/zabbix/.ssh, что соответствует параметру конфигурации SSHKeyLocation сервера Zabbix.

Attention:
Типы ключей, отличные от "rsa", могут поддерживаться утилитой ssh-keygen и SSH-серверами, но могут не поддерживаться libssh2, используемой Zabbix.

Форма настройки shell

Этот шаг следует выполнить только один раз для каждого узла сети, который будет контролироваться с помощью проверок SSH.

С помощью следующих команд файл **открытого** ключа можно установить на удаленный узел сети *10.10.10.10*, чтобы проверки SSH можно было выполнять с учетной записью *root* (для лучшей читаемости приглашения командной строки опущены):

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

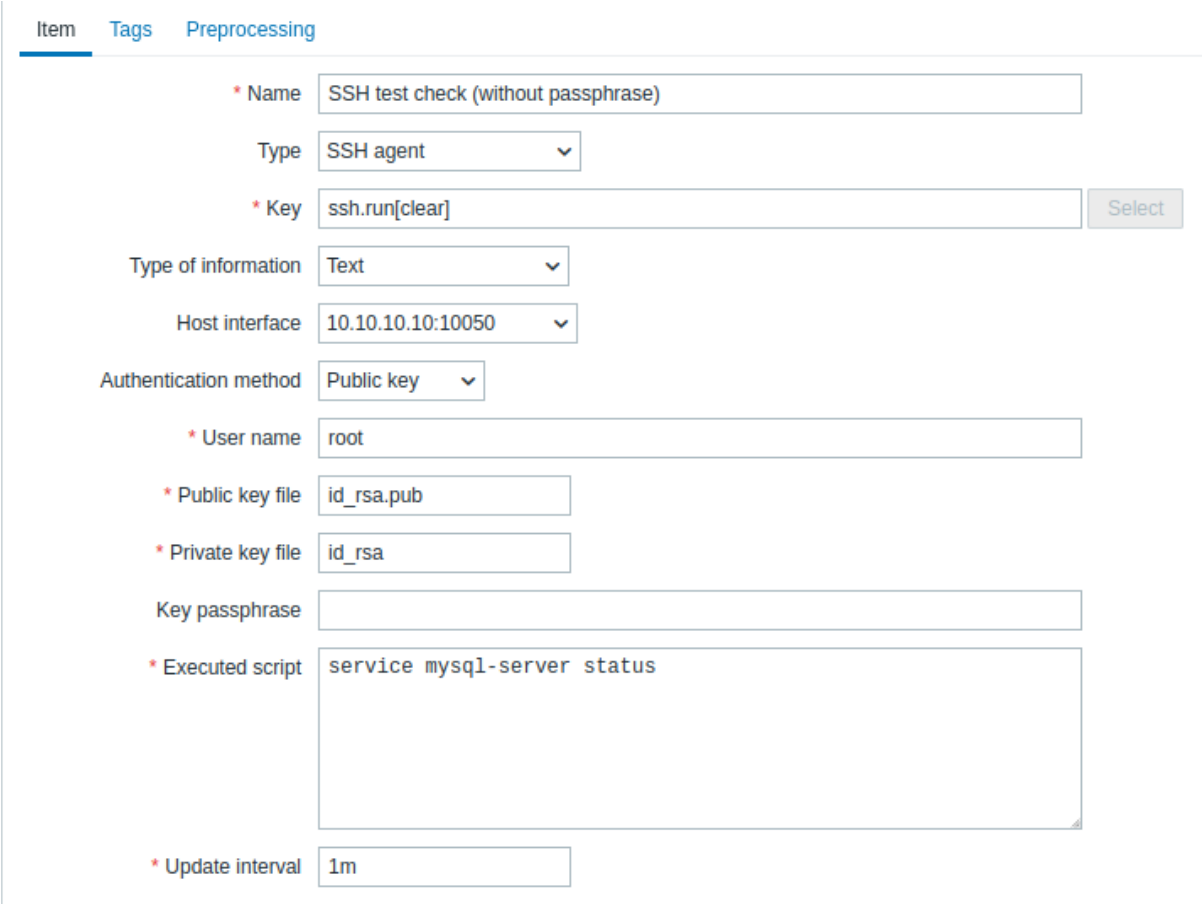
Теперь можно проверить вход по SSH, используя ключ по умолчанию (/home/zabbix/.ssh/id_rsa) для учетной записи пользователя *zabbix*:

```
sudo -u zabbix ssh root@10.10.10.10
```

Если вход выполнен успешно, настройка shell завершена, и удаленный сеанс SSH можно закрыть.

Настройка элемента данных

Фактические команды, которые необходимо выполнить, следует указать в поле *Выполняемый скрипт* в настройках элемента данных. Несколько команд можно выполнить последовательно, разместив их на новых строках. В этом случае возвращаемые значения также будут отформатированы в несколько строк.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие указания особых параметров для элементов данных SSH:

Параметр	Описание	Комментарии
Тип	Выберите здесь **SSH-агент**.	

Параметр	Описание	Комментарии

Ключ	Уникальный (для каждого узла сети) ключ элемента данных в формате `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	Параметр **unique short description** является обязательным и должен быть уникальным для каждого элемента данных SSH на узле сети.
		
		Порт по умолчанию — 22, а не порт, указанный в интерфейсе, которому назначен этот элемент данных.
		
		Параметр **ssh options** позволяет передавать дополнительные параметры SSH в формате `key1=value1;key2=value2,value3`. Несколько значений для одного ключа можно передать через запятую (в этом случае параметр необходимо заключить в кавычки); несколько ключей параметров можно передать через точку с запятой.
		
		Поддерживаются следующие ключи параметров: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Поддержка ключей и значений параметров зависит от библиотеки SSH (например, `PubkeyAcceptedKeyTypes` поддерживается только в libssh); если параметр не поддерживается, будет возвращена ошибка, а элемент данных станет неподдерживаемым.
		
		Обратите внимание, что знак "+" для добавления настроек шифров и знак "!" для отключения определенных настроек шифров (как в GnuTLS и OpenSSL) не поддерживаются.
		
		Примеры:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-h`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubke`
		
		Параметр **subsystem** позволяет передать подсистему SSH, ограничивая SSH-подключение определенными операциями, разрешенными этой подсистемой (например, передачей файлов с использованием SFTP или управлением сетевыми устройствами с использованием NETCONF). Обратите внимание, что использование подсистемы также может потребовать применения специального синтаксиса скрипта в параметре *Выполняемый скрипт*.
		
		Примеры:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Метод аутентификации	Один из вариантов: «Пароль» или «Открытый ключ».	

Параметр	Описание	Комментарии
Имя пользователя	Имя пользователя (до 255 символов) для аутентификации на удаленном узле сети. Обязательное поле.	
Файл открытого ключа	Имя файла открытого ключа, если в поле *Метод аутентификации* выбран вариант «Открытый ключ». Обязательное поле.	Пример: *id_rsa.pub* — имя файла открытого ключа по умолчанию, созданного командой ssh-keygen.
Файл закрытого ключа	Имя файла закрытого ключа, если в поле *Метод аутентификации* выбран вариант «Открытый ключ». Обязательное поле.	Пример: *id_rsa* — имя файла закрытого ключа по умолчанию.
Пароль или *Парольная фраза ключа*	Пароль (до 255 символов) для аутентификации или парольная фраза, **если** она использовалась для закрытого ключа.	Оставьте поле *Парольная фраза ключа* пустым, если парольная фраза не использовалась. См. также информацию об известных проблемах, связанных с использованием парольной фразы.

Выполняемый скрипт	Команда или команды оболочки, выполняемые через удаленный сеанс SSH.	Возвращаемое значение выполненной команды или команд оболочки ограничено 16 МБ (включая конечные пробелы, которые усекаются); также применяются ограничения базы данных.
		
		Обратите внимание, что библиотека libssh2 может обрезать исполняемые скрипты до ~32 КБ.
		
		Примеры:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Пример (для подсистемы NETCONF):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Telnet агент

Обзор

Telnet проверки выполняются без наличия какого-либо агента. Zabbix агент не требуется для проверок через Telnet.

Настраиваемые поля

Фактические команда(ы) для выполнения необходимо поместить в поле **Выполняемый скрипт** при настройке элемента данных.
 Можно выполнять несколько команд одну за другой, размещая каждую на новой строке. В этом случае значение результата также будет отформатировано как многострочное.

Поддерживаемые символы, которыми может заканчиваться строка приглашения командной оболочки:

- $
- #
- >
- %

Note:
Строка telnet с запросом, которая заканчивается одним из этих символов, будет удалена из возвращаемого значения, но только для первой команды из списка команд, то есть, только в начале telnet сессии.

Ключ	Описание
telnet.run[<уникальное короткое описание>,<ip>,<порт>,<кодировка>]	Выполнение команды на удалённом устройстве с использованием telnet соединения

Attention:
Если telnet проверка возвращает значение с не-ASCII символами и не в UTF8 кодировке, то необходимо указать корректный параметр ключа *<кодировка>*. Смотрите страницу о кодировке возвращаемых значений для получения более подробных сведений.

JMX агент

Обзор

Мониторинг JMX можно использовать для мониторинга счётчиков JMX в Java приложениях.

Мониторинг JMX имеет в Zabbix встроенную поддержку в виде демона Zabbix, называемого «Zabbix Java gateway».

Когда Zabbix сервер хочет узнать значение конкретного счетчика JMX у узла сети, он опрашивает Zabbix **Java gateway**, который, в свою очередь, использует API управления JMX [en], чтобы удалённо опросить интересующее приложение.

Для получения более подробных сведений, включая настройку, смотрите раздел Zabbix Java gateway.

Warning:
Подключения между Java gateway и наблюдаемым JMX приложением не должны быть закрыты брандмауэром.

Включение удалённого JMX мониторинга для Java приложений

Приложению Java не требуется какое-либо дополнительно установленное программное обеспечение, но для поддержки удалённого мониторинга JMX приложение необходимо запустить с указанными ниже параметрами командной строки.

Как минимум, если вы просто хотите начать мониторинг простого приложения Java на локальном хосте без какой-либо защиты, запустите его со следующими параметрами:

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.registry.ssl=false \
-jar /    /   /      .jar
```

С этими аргументами Java будет слушать входящие соединения JMX на порту 12345, только с локального хоста, без необходимости в аутентификации или SSL.

Если вы хотите разрешить подключения с другого интерфейса, укажите параметр -Djava.rmi.server.hostname равным IP адресу этого интерфейса.

Если вы хотите иметь более строгую проверку в плане безопасности, вам доступно множество других опций Java. Скажем, следующий пример запускает приложение с более мудрёным набором опций и открывает это приложение для более широкой сети, а не только для локального компьютера.

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$  _     _     \
-Djavax.net.ssl.keyStorePassword=$  _    _ _     _     \
-Djavax.net.ssl.trustStore=$  _     _      \
-Djavax.net.ssl.trustStorePassword=$  _    _ _      _      \
```

```
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /   /   /      .jar
```

Большинство (если не все) из этих настроек можно указать в `$JRE/lib/management/management.properties` (или там, где этот файл расположен на вашем компьютере).

Обратите внимание, что если вы хотите использовать SSL, вам необходимо изменить скрипт startup.sh, добавив в него опции Java gateway `-Djavax.net.ssl.*`, чтобы он знал, где найти хранилище ключей и доверенное хранилище.

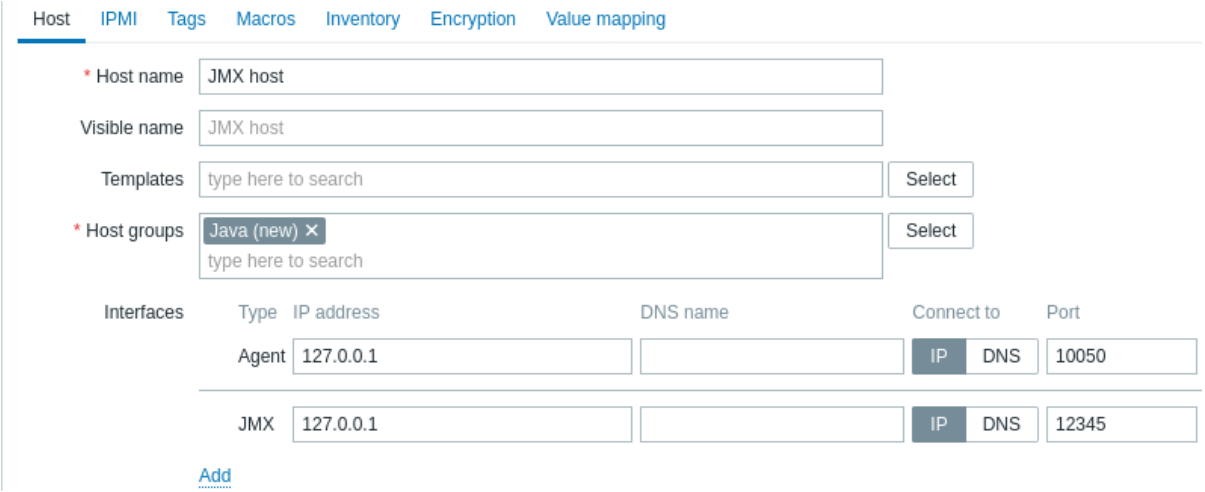
Смотрите Мониторинг и Управление с использованием JMX [en] для получения более подробной информации.

Настройка JMX интерфейсов и элементов данных в веб-интерфейсе Zabbix

Когда Java Gateway запущен, сервер знает, где его искать, и Java приложение запущено с поддержкой удалённого JMX мониторинга, самое время настроить интерфейсы и элементы данных в веб-интерфейсе Zabbix.

Настройка JMX интерфейса

Начнём с создания интерфейса типа JMX у интересующего узла сети.

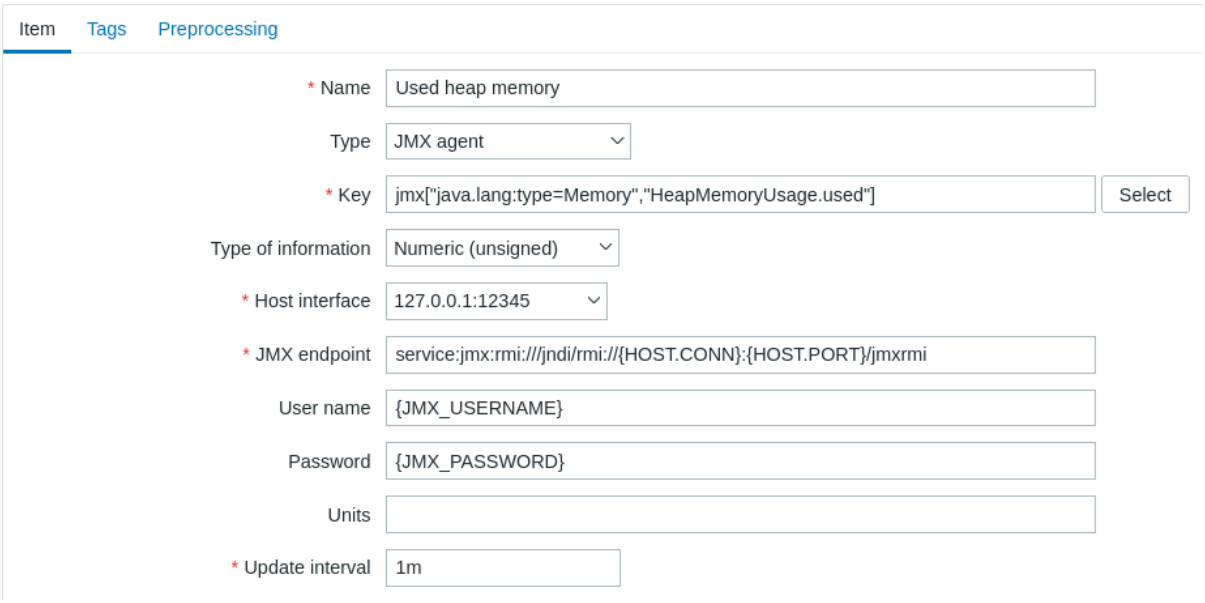


Все обязательные поля ввода отмечены красной звёздочкой.

Добавление элемента данных JMX agent

Для каждого счетчика JMX, который вас интересует, добавьте элемент данных **JMX agent**, привязанный к этому интерфейсу.

Ключ на скриншоте ниже: `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, для которых требуется конкретная информация для элементов данных JMX:

Type	Укажите здесь **JMX agent**.

Key	Ключ элемента данных `jmx[]` содержит три параметра: **object name** - имя объекта MBean **attribute name** - имя атрибута MBean с необязательными именами полей составных данных, разделенными точками **unique short description** - уникальное описание, которое позволяет использовать несколько элементов данных JMX с одинаковыми object name и attribute name на узле сети (необязательно) Подробнее о ключах элементов данных JMX см. ниже. Вы можете обнаруживать MBean и атрибуты MBean с помощью элемента данных `jmx.discovery[]` low-level discovery.
JMX endpoint	Вы можете указать пользовательский JMX endpoint. Убедитесь, что параметры подключения JMX endpoint соответствуют интерфейсу JMX. Это можно сделать с помощью макросов `{HOST.*}`, как в стандартном JMX endpoint. Поддерживаются макросы `{HOST.*}` и пользовательские макросы.
User name	Укажите имя пользователя (до 255 символов), если вы настроили аутентификацию в вашем Java-приложении. Поддерживаются пользовательские макросы.
Password	Укажите пароль (до 255 символов), если вы настроили аутентификацию в вашем Java-приложении. Поддерживаются пользовательские макросы.

Если вы хотите отслеживать логический счетчик, который принимает значение ”true” или ”false”, укажите тип информации как ”Numeric (unsigned)” и выберите шаг предварительной обработки ”Boolean to decimal” на вкладке Preprocessing. Сервер будет сохранять логические значения как 1 или 0 соответственно.

Более подробно о ключах элементов данных JMX

Простые атрибуты

Имя объекта MBean — это не что иное, как строка, которую вы определили в вашем Java приложении. Имя атрибута, с другой стороны, может быть более сложным. В случае, если атрибут возвращает простой тип данных (целое число, строку и т.п.), то беспокоиться не о чем, ключ будет выглядеть примерно так:

```
jmx[com.example:Type=Hello,weight]
```

В этом примере именем объекта является «com.example:Type=Hello», именем атрибута будет «weight», а тип возвращаемого значения, скорее всего, должен быть «Числовой (с плавающей точкой)».

Атрибуты, возвращающие составные данные

Ключ становится более сложным, когда ваш атрибут возвращает составные данные. Например: именем вашего атрибута является «apple», и он возвращает хеш представляющих его параметров, таких как «weight», «color» и прочее. Тогда ваш ключ может выглядеть примерно так:

```
jmx[com.example:Type=Hello,apple.weight]
```

Этот пример показывает, как разделяются с помощью точки имя атрибута и ключ хеша. Точно также, если атрибут возвращает вложенные составные данные, их части снова нужно разделять точкой:

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

Атрибуты, возвращающие табличные данные

Атрибуты с табличными данными состоят из одного или нескольких составных атрибутов. Если такой атрибут указан в параметре имени атрибута, то значение этого элемента данных вернёт всю структуру атрибута в формате JSON. Значения отдельных элементов из табличных данных можно извлечь с помощью предобработки.

Пример атрибута с табличными данными:

```
jmx[com.example:type=Hello,foodinfo]
```

Значение элемента данных:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
```

```
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

Проблема с точками

Пока все хорошо. Но что, если имя атрибута или ключ хеша содержит символ точки? Вот пример:

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

Это проблема. Как сказать Zabbix'у, что имя атрибута — «all.fruits», а не просто «all»? Как отличить точку, которая является частью имени, от точки, которая разделяет имя атрибута и ключи хешей?

Это возможно; всё, что вам требуется сделать — экранировать точки, которые являются частью имени, обратной косой чертой:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Аналогично, если ваш ключ хеша содержит точку, вам необходимо её экранировать:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

Другие проблемы

Символ обратной косой черты в имени атрибута также необходимо экранировать:

```
jmx[com.example:type=Hello,c:\\documents]
```

Для обработки любых других символов в ключе JMX элемента данных, пожалуйста, обратитесь к разделу формата ключа элементов данных.

На самом деле это всё, что нужно сделать. Успешного мониторинга JMX!

Непримитивные типы данных

Имеется возможность работы с пользовательскими MBean, возвращающими непримитивные типы данных, которые переопределяют метод **toString()**.

Использование пользовательской конечной точки с JBoss EAP 6.4

Пользовательские конечные точки позволяют работать с различными транспортными протоколами, отличными от протокола RMI по умолчанию.

Чтобы продемонстрировать эту возможность, попробуем настроить мониторинг JBoss EAP 6.4 в качестве примера. Сначала сделаем несколько допущений:

- У вас уже установлен Zabbix Java gateway. Если нет, вы можете сделать это в соответствии с документацией.
- Сервер Zabbix и Java gateway установлены с префиксом `/usr/local/`
- JBoss уже установлен в `/opt/jboss-eap-6.4/` и работает в автономном режиме
- Предположим, что все эти компоненты работают на одном узле сети
- Брандмауэр и SELinux отключены (или настроены соответствующим образом)

Сделаем несколько простых настроек в `zabbix_server.conf`:

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

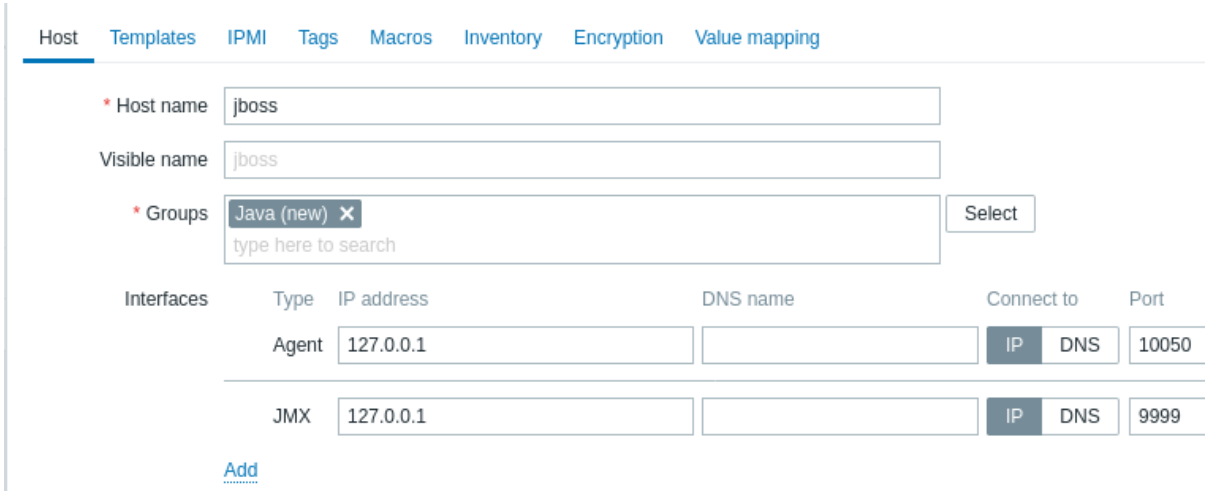
И в конфигурационном файле `zabbix_java/settings.sh` (или `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Проверьте, что JBoss прослушивает свой стандартный порт управления:

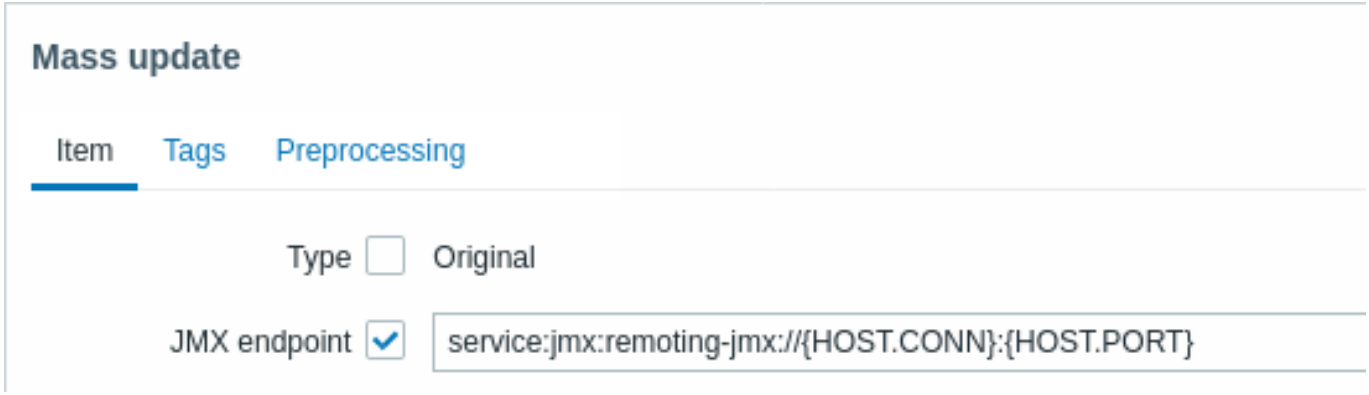
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Теперь создадим узел сети с интерфейсом JMX 127.0.0.1:9999 в Zabbix.



Поскольку мы знаем, что эта версия JBoss использует протокол JBoss Remoting вместо RMI, мы можем массово обновить параметр конечной точки JMX для элементов данных в нашем шаблоне JMX соответствующим образом:

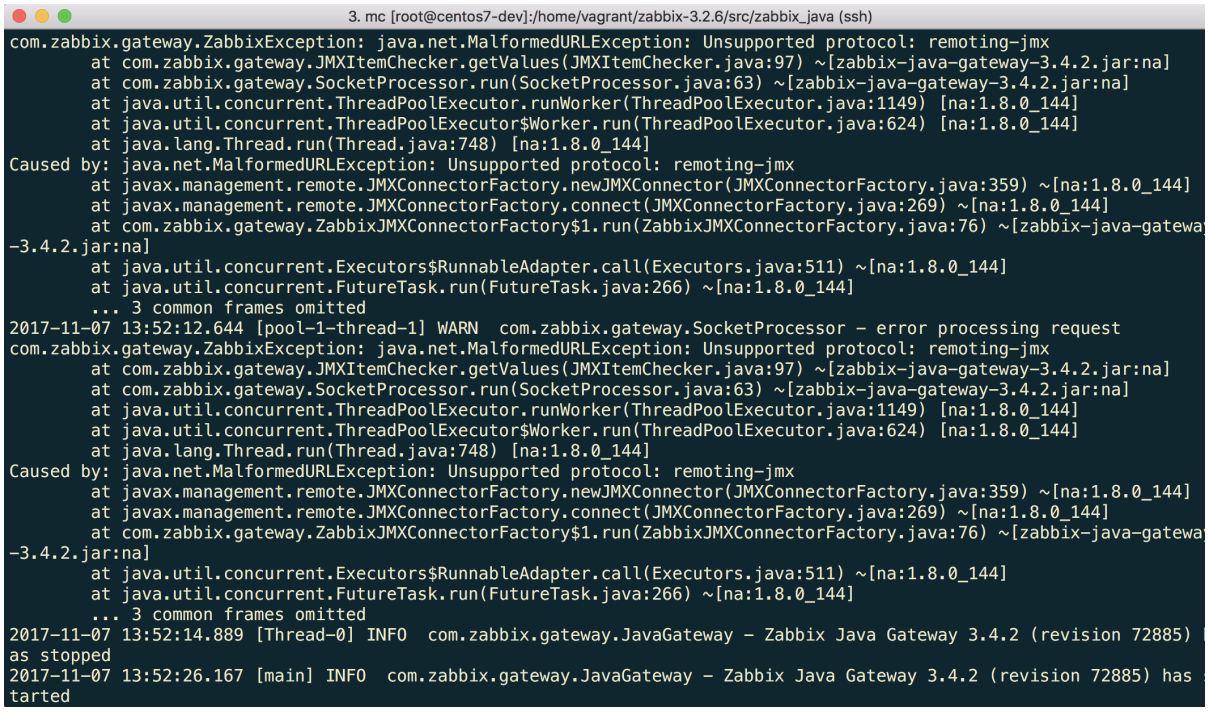
`service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}`



Обновим кэш конфигурации:

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

Обратите внимание, что сначала вы можете столкнуться с ошибкой.



"Unsupported protocol: remoting-jmx" означает, что Java gateway не знает, как работать с указанным протоколом. Это можно исправить, создав файл `~/needed_modules.txt` со следующим содержимым:

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

а затем выполнив команду:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Таким образом, Java gateway будет иметь все необходимые модули для работы с jmx-remoting. Остается перезапустить Java gateway, немного подождать и, если вы все сделали правильно, увидеть, что данные мониторинга JMX начинают поступать в Zabbix (см. также: Latest data).

Вычисляемый элемент данных

Обзор

Этот тип элемента данных позволяет выполнять вычисления на основе уже существующих метрик. Вычисляемый элемент данных не собирает данные самостоятельно.

Например, можно:

- Сложить два значения.
- Вычислить среднее значение нескольких элементов данных.
- Вычислить общее значение для группы элементов данных.

Результирующее значение сохраняется в базе данных Zabbix так же, как и для любого другого элемента данных; сохраняются как значения истории, так и значения трендов, а также могут создаваться графики.

Вычисление выполняется на основе формулы. Синтаксис формулы совпадает с синтаксисом выражений триггеров.

Фактически, можно упростить триггеры, разместив сложную логику обработки данных в вычисляемом элементе данных, а затем ссылаться на вычисляемый элемент данных в триггере с помощью ключа элемента данных.

Существует два способа указать источник данных для вычислений:

1. Указать элементы данных в формуле.
2. Выполнить агрегацию данных на основе фильтра, выбирающего элементы данных.

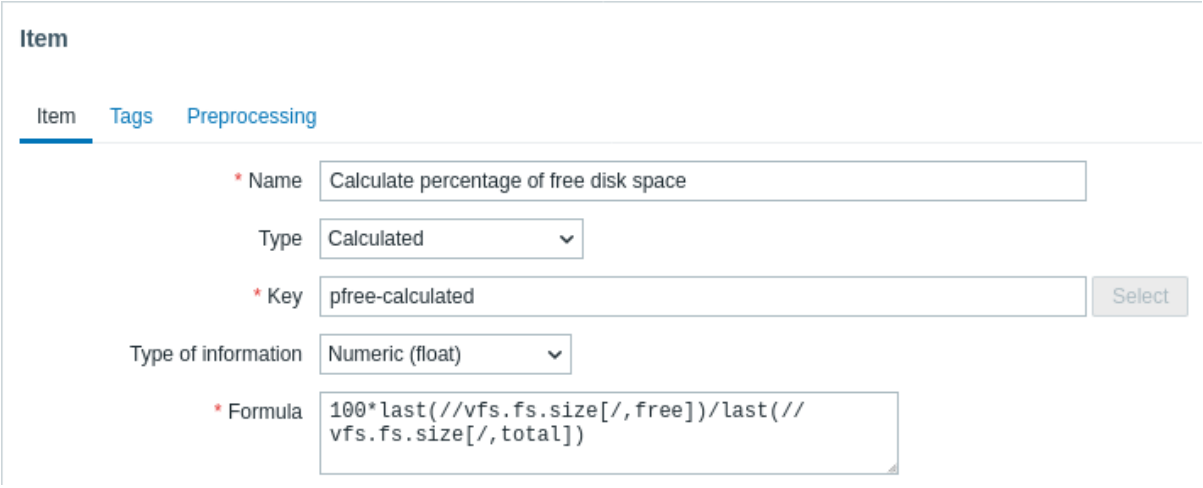
Второй вариант позволяет вычислять средние значения для групп, общие значения и т. д.

Все вычисления выполняются сервером Zabbix и обрабатываются процессами `history poller`; прокси и агенты не участвуют в этом процессе. Обратите внимание, что активное использование вычисляемых элементов данных с большими временными периодами может повлиять на производительность сервера.

Обратите внимание, что вычисляемые элементы данных могут получать доступ к истории других элементов данных, включая элементы данных, к которым у пользователя, создающего вычисляемый элемент данных, нет доступа. Если такое поведение нежелательно, вычисляемые элементы данных можно отключить, установив значение конфигурационного параметра `StartHistoryPollers` равным 0.

Настраиваемые поля

Чтобы использовать вычисляемые элементы данных, выберите тип элемента данных **Вычисляемый**.



Ключ — это уникальный идентификатор элемента данных (для каждого узла сети). Вы можете создать любое имя ключа, используя поддерживаемые символы. Ключ позволяет ссылаться на вычисляемый элемент данных в триггерах.

Определение вычисления необходимо ввести в поле **Формула**.

Между формулой и ключом нет связи. Параметры ключа никак не используются в формуле.

В отличие от выражений триггеров, сервер Zabbix обрабатывает вычисляемые элементы данных в соответствии с интервалом обновления элемента данных, а не при получении нового значения.

Обратите внимание: если в качестве типа информации выбран *Числовой (без знака)*, а результат вычисления имеет тип числа с плавающей запятой, он усекается до целого числа.

Формула простого вычисления

Синтаксис простой формулы:

`function(/host/key,<parameter1>,<parameter2>,...)`

где:

Элемент	Описание
`function`	Одна из поддерживаемых функций: last, min, max, avg, count и т. д.
`host`	Узел сети элемента данных, который используется для вычисления. Текущий узел сети можно не указывать (то есть как в `function(//key,parameter,...)`).
`key`	Ключ элемента данных, который используется для вычисления. Элементы данных, возвращающие значения с типом данных binary или JSON, не поддерживаются. Если вы измените ключ элемента данных у элемента данных, на который есть ссылка, ключ необходимо будет вручную обновить в формуле. Все элементы данных, на которые ссылаются функции истории, должны существовать и собирать данные.
`parameter(s)`	Параметры функции, если требуются. Поддерживаются суффиксы времени и суффиксы размера памяти.

Более сложная формула может использовать комбинацию функций, операторов и скобок.

См. примеры формул вычисляемых элементов данных:

- (1) Общий трафик на интерфейсе
- (2) Процент свободного места на диске
- (3) Процент входящего трафика
- (4) Среднее количество обработанных значений

Некоторые переменные могут передаваться в формулу с помощью пользовательских макросов. Обратите внимание, что пользовательские макросы:

- будут разрешены — если используются для ссылки на параметр функции, параметр фильтра элемента данных или константу.
- **не будут** разрешены — если ссылаются на функцию, имя узла сети, ключ элемента данных, параметр ключа элемента данных или оператор.

Допускается сравнение со строками.

Если в кэше нет недавних данных и в функции не определен период запроса, Zabbix по умолчанию будет запрашивать исторические значения из базы данных, уходя в прошлое максимум на одну неделю.

Вычисляемый элемент данных может стать неподдерживаемым в нескольких случаях:

- элемент данных, на который есть ссылка,
  - не найден
  - отключен или принадлежит отключенному узлу сети
  - не поддерживается (кроме функции nodata() и операторов с неизвестными значениями)
- нет данных для вычисления функции
- деление на ноль
- используется некорректный синтаксис

Агрегация данных

Агрегация данных основана на фильтре, который выбирает несколько элементов данных. Агрегация данных также является типом вычисляемого элемента данных, однако в формуле используется немного другой синтаксис.

Чтобы получить агрегированные значения, используйте одну из поддерживаемых агрегирующих функций: `avg`, `max`, `min`, `sum` и т. д. Затем добавьте функцию **foreach** как единственный параметр и её фильтр элементов данных для выбора нужных элементов данных:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Функция **foreach** (например, *avg_foreach*, *count_foreach* и т. д.) возвращает одно агрегированное значение для каждого выбранного элемента данных. Элементы данных выбираются с помощью фильтра элементов данных (`/host/key?[group="host group"]`) из истории элементов данных. Подробнее см. в разделе функции foreach.

Если у некоторых элементов данных нет данных за запрошенный период, они игнорируются при вычислении. Если данных нет ни у одного элемента данных, функция вернёт ошибку.

В качестве альтернативы можно перечислить несколько элементов данных как параметры для агрегации:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Обратите внимание, что `function` здесь должна быть функцией history/trend.

См. примеры агрегирующих вычислений:

- (5) Общий входящий трафик для узла сети
- (6) Общее дисковое пространство для группы
- (7) Средняя загрузка процессора для группы
- (8) Средняя загрузка процессора для группы и тега
- (9) Среднее количество обработанных значений
- (10) Среднее количество запросов для группы
- (11) Общее количество успешных DNS-проверок
- (12) Общее количество неподдерживаемых элементов данных для группы
- (13) Статистика интерфейса для группы

Пользовательские макросы и макросы low-level discovery поддерживаются в:

- параметрах ключа элемента данных
- параметрах функции
- условиях фильтра (имя группы узлов сети и имя тега)
- константах выражения

Агрегирующее вычисление может стать неподдерживаемым, если:

- не найден ни один из указанных элементов данных (это может произойти, если ключ элемента данных указан неверно, ни один из элементов данных не существует или все включённые группы указаны неверно)
- отсутствуют данные для вычисления функции

Примеры вычисляемых элементов данных

(1) Общий трафик на интерфейсе

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Вычисляет общую пропускную способность на eth0.

(2) Процент свободного дискового пространства

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix возьмет последние значения свободного и общего дискового пространства (на /) и вычислит процент по заданной формуле.

(3) Процент входящего трафика

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Рассчитайте процент входящего трафика от общего трафика.

(4) Среднее количество обработанных значений

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Вычисляет 10-минутное среднее количество значений, обработанных Zabbix.

(5) Общий входящий трафик для узла сети

```
sum(last_foreach(/host/net.if.in[*]))
```

Вычисляет сумму всех элементов данных, соответствующих `net.if.in[*]` на узле сети.

(6) Общее дисковое пространство для группы

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Вычислить общее дисковое пространство для всех `MySQL Servers`.

(7) Средняя загрузка процессора для группы

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Вычислить среднюю загрузку процессора для всех `MySQL Servers`.

(8) Средняя загрузка процессора для группы и тега

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Вычислить среднюю загрузку CPU на всех узлах сети в нескольких группах узлов сети, имеющих определенные теги.

(9) Среднее количество запросов для группы

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Вычисляет 5-минутное среднее количества запросов в секунду для `MySQL Servers`.

(10) Общее количество успешных DNS-проверок

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Вычисляет общее количество успешных DNS-проверок на всех узлах сети. Показанная форма элемента данных приведена в качестве примера: `net.dns[192.0.2.1,example.com,A]` — это один из возможных ключей.

Обратите внимание, что подстановочные знаки должны соответствовать количеству параметров в ключе (здесь `net.dns` имеет три параметра: ip, name, type).

(11) Общее количество неподдерживаемых элементов данных для группы

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Вычисляет общее количество неподдерживаемых элементов данных для `Zabbix servers`.

(12) Статистика интерфейса для группы

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Вычислить статистику интерфейса для группы `video`.

Примеры правильного и неправильного синтаксиса

Выражения (включая вызовы функций) нельзя использовать в качестве параметров функций history, trend или foreach. Однако сами эти функции можно использовать в параметрах других (не исторических) функций.

Выражение	Пример
Допустимо	`avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)`
	`max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))`
Недопустимо	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Обратите внимание, что в таком выражении:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(где `resptime[*]` — пример шаблона ключа элемента данных, соответствующего ключам `resptime[1]`, `resptime[2]` и т. д., который используется для получения времени отклика от нескольких активных узлов кластера) нельзя гарантировать, что обе части уравнения всегда будут иметь одинаковый набор значений. Пока вычисляется одна часть выражения, может поступить новое значение за запрошенный период, и тогда другая часть выражения будет иметь другой набор значений.

Зависимый элемент данных

Обзор

Существуют ситуации, когда один элемент данных собирает несколько метрик одновременно или когда даже более целесообразно собирать связанные метрики одновременно, например:

- загрузка CPU отдельных ядер
- входящий/исходящий/общий сетевой трафик

Чтобы обеспечить массовый сбор метрик и одновременное использование в нескольких связанных элементах данных, Zabbix поддерживает зависимые элементы данных. Зависимые элементы данных зависят от мастер-элемента, который собирает их данные одновременно, одним запросом. Новое значение мастер-элемента автоматически заполняет значения зависимых элементов данных. Зависимые элементы данных не могут иметь интервал обновления, отличный от интервала мастер-элемента.

Параметры предварительной обработки Zabbix можно использовать для извлечения из данных мастер-элемента той части, которая нужна для зависимого элемента данных.

Предварительная обработка управляется процессом `preprocessing manager` вместе с потоками-исполнителями, которые выполняют шаги предварительной обработки. Все значения с предварительной обработкой, полученные от разных сборщиков данных, проходят через `preprocessing manager` перед добавлением в кэш истории. Для связи между сборщиками данных (pollers, trappers и т. д.) и процессом предварительной обработки используется IPC-связь на основе сокетов.

Сервер Zabbix или прокси Zabbix (если узел сети контролируется через прокси) выполняют шаги предварительной обработки и обработку зависимых элементов данных.

В качестве мастер-элементов можно задавать элементы данных любого типа, включая зависимые элементы данных. Можно использовать дополнительные уровни зависимых элементов данных, чтобы извлекать более мелкие части из значения существующего зависимого элемента данных.

Ограничения

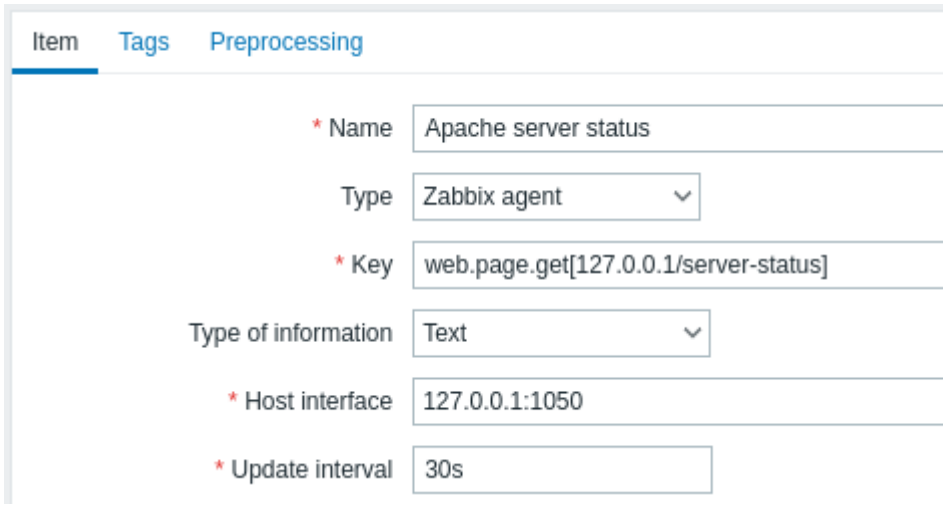
- Разрешены только зависимости в пределах одного узла сети (шаблона)
- Прототип элемента данных может зависеть от другого прототипа элемента данных или обычного элемента данных на том же узле сети
- Зависимый элемент данных на узле сети с мастер-элементом из шаблона не будет экспортирован в XML

Настройка элемента данных

Зависимый элемент данных зависит от своего мастер-элемента в части данных. Поэтому сначала необходимо настроить (или чтобы уже существовал) **мастер-элемент**:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Нажмите *Элементы данных* в строке узла сети.

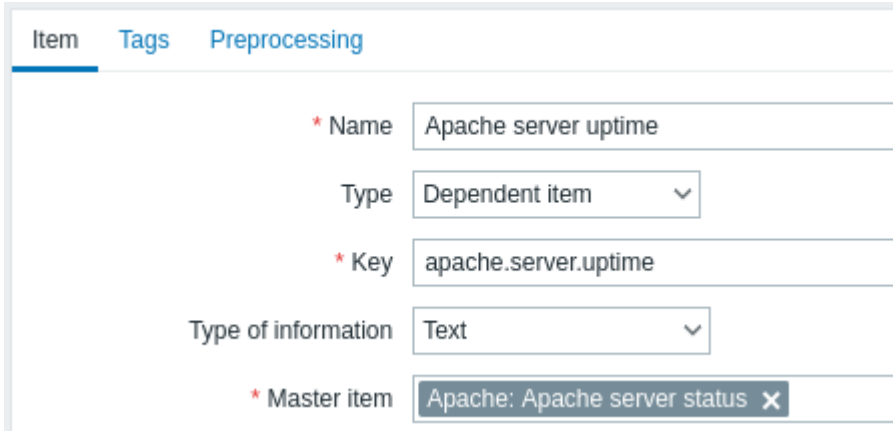
3. Нажмите *Создать элемент данных*.
4. Введите параметры элемента данных в форме.



Все обязательные поля ввода отмечены красной звездочкой.

Нажмите *Добавить*, чтобы сохранить мастер-элемент.

Затем можно настроить **зависимый элемент данных**.

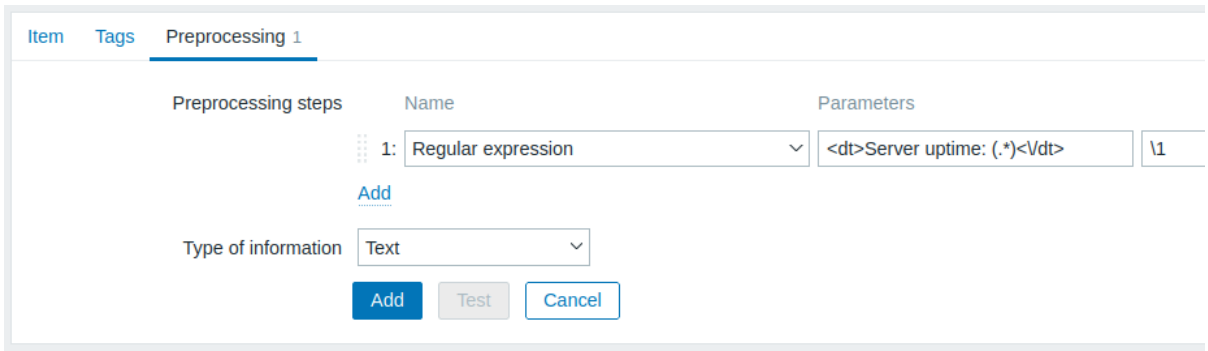


Все обязательные поля ввода отмечены красной звездочкой.

Поля, для которых требуется особая информация для зависимых элементов данных:

Тип	Выберите здесь **Зависимый элемент данных**.
Ключ	Введите ключ, который будет использоваться для идентификации элемента данных.
Мастер-элемент	Выберите мастер-элемент. Значение мастер-элемента будет использоваться для заполнения значения зависимого элемента данных.
Тип информации	Выберите тип информации, который будет соответствовать формату данных, которые будут храниться.

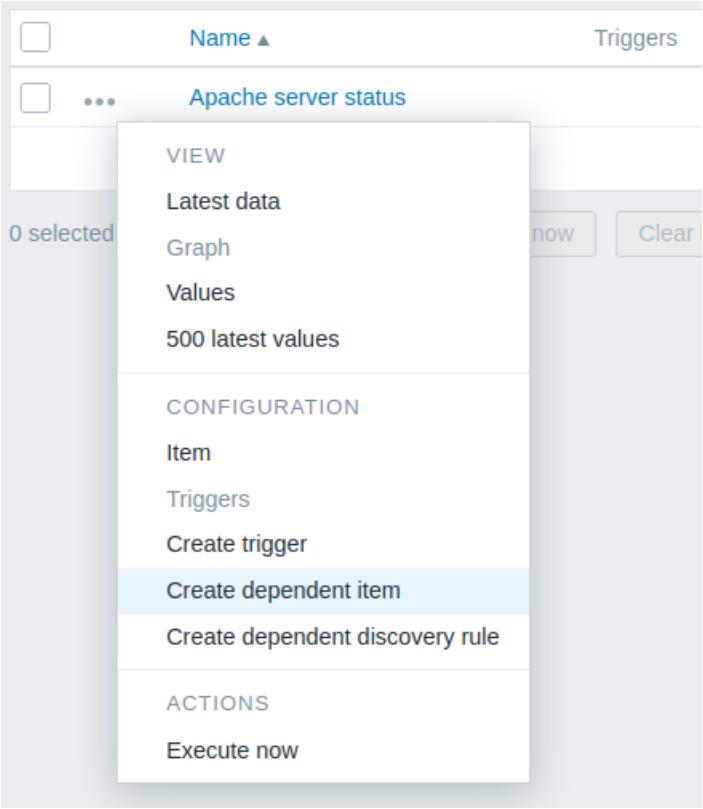
Вы можете использовать предобработку значения элемента данных, чтобы извлечь нужную часть значения мастер-элемента.



Без предобработки значение зависимого элемента данных будет точно таким же, как значение мастер-элемента.

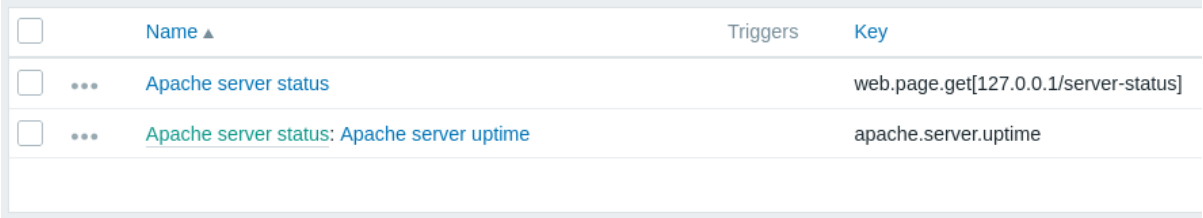
Нажмите *Добавить*, чтобы сохранить зависимый элемент данных.

Быстрый способ создания зависимого элемента данных доступен при нажатии кнопки ••• в списке элементов данных и выборе *Создать зависимый элемент данных*.



Отображение

В списке элементов данных зависимые элементы данных отображаются с именем основного элемента данных в качестве префикса.



Если основной элемент данных удаляется, то будут удалены и все его зависимые элементы данных.

Элемент данных типа скрипт

Обзор

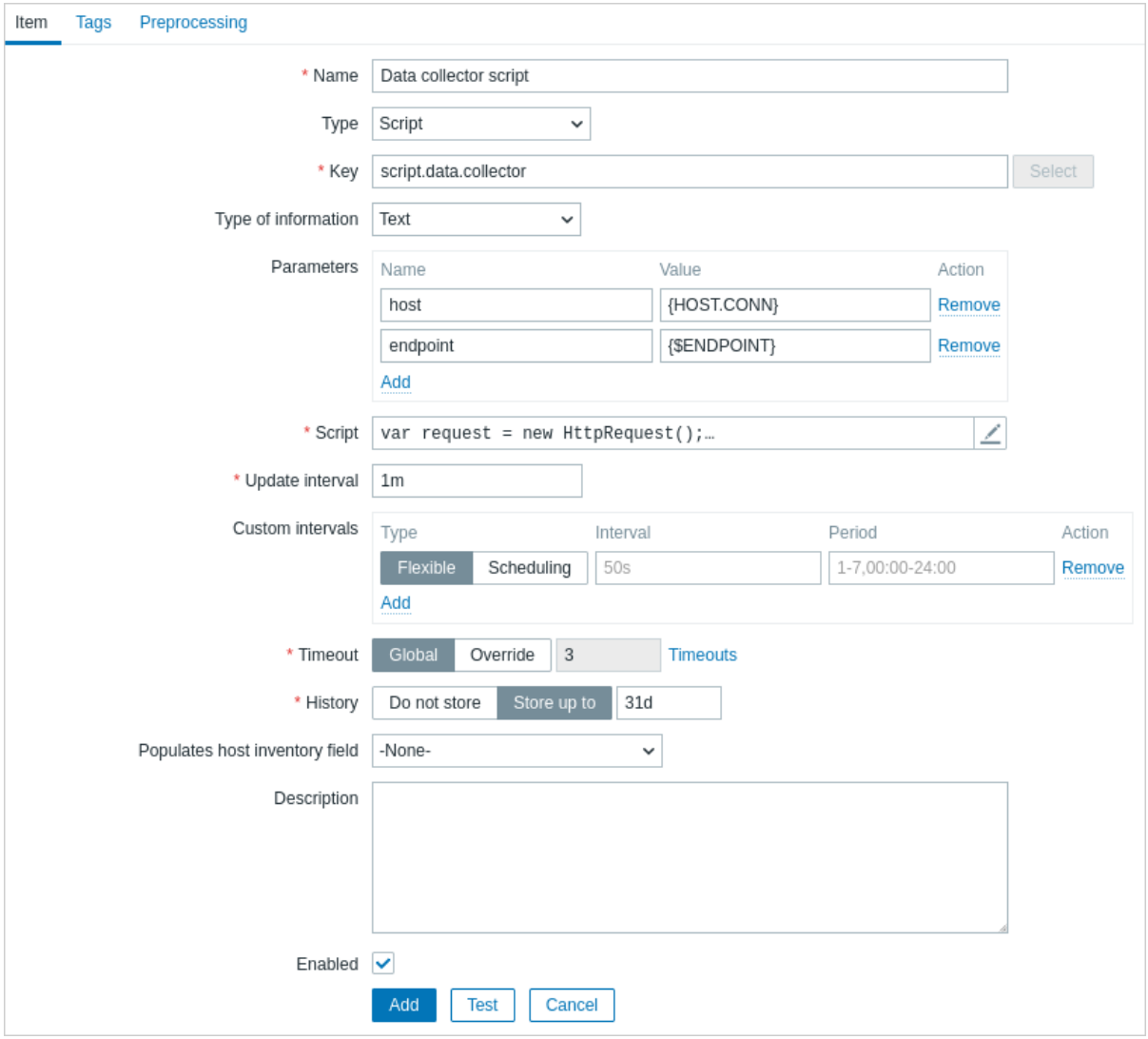
Скриптовые элементы данных можно использовать для сбора данных путём выполнения пользовательского JavaScript кода с возможностью получения данных через HTTP/HTTPS. В дополнение к скрипту можно указать опциональный список параметров (пары из имени и значения) и время ожидания.

Этот тип элементов данных может быть полезен в сценариях сбора данных, требующих нескольких шагов или сложной логики. Например, скриптовый элемент данных можно настроить на выполнение HTTP вызова, последующей обработки неким образом данных, полученных на первом шаге, и передачи преобразованного значения второму HTTP вызову.

Скриптовые элементы данные обрабатываются Zabbix сервером или прокси на уровне поллеров.

Настройка

В поле *Тип* диалога настройки элемента данных выберите Скрипт, затем заполните необходимые поля.



Все обязательные поля ввода отмечены красной звёздочкой.

Поля, которые требуют информацию, специфичную для элементов данных Скрипт:

Поле	Описание
Ключ (Key)	Введите уникальный ключ, который будет использоваться для идентификации этого элемента данных.
Параметры (Parameters)	Укажите переменные, которые будут переданы скрипту как пары атрибутов и значений. Поддерживаются пользовательские макросы. Чтобы узнать, какие встроенные макросы поддерживаются, выполните поиск по запросу «Скриптовый элемент данных» в таблице поддерживаемых макросов.
Скрипт (Script)	Введите код JavaScript в модальном редакторе, который открывается при нажатии на поле параметров или на кнопку со значком карандаша сразу за полем параметров. Этот код должен предоставлять логику для получения значения метрики. Код имеет доступ ко всем параметрам и всем дополнительным объектам JavaScript, добавленным Zabbix. Смотрите также: Руководство пользователя JavaScript.
Время ожидания (Timeout)	Время ожидания выполнения JavaScript (1-600s, превышение этого значения вернёт ошибку). Обратите внимание, что в зависимости от скрипта для срабатывания тайм-аута может потребоваться больше времени. Дополнительную информацию о параметре *Timeout* смотрите в разделе общие атрибуты элементов данных.

Примеры

Простой сбор данных

Получение содержимого *https://www.example.com/release_notes*:

- Создайте элемент данных с типом «Скрипт».

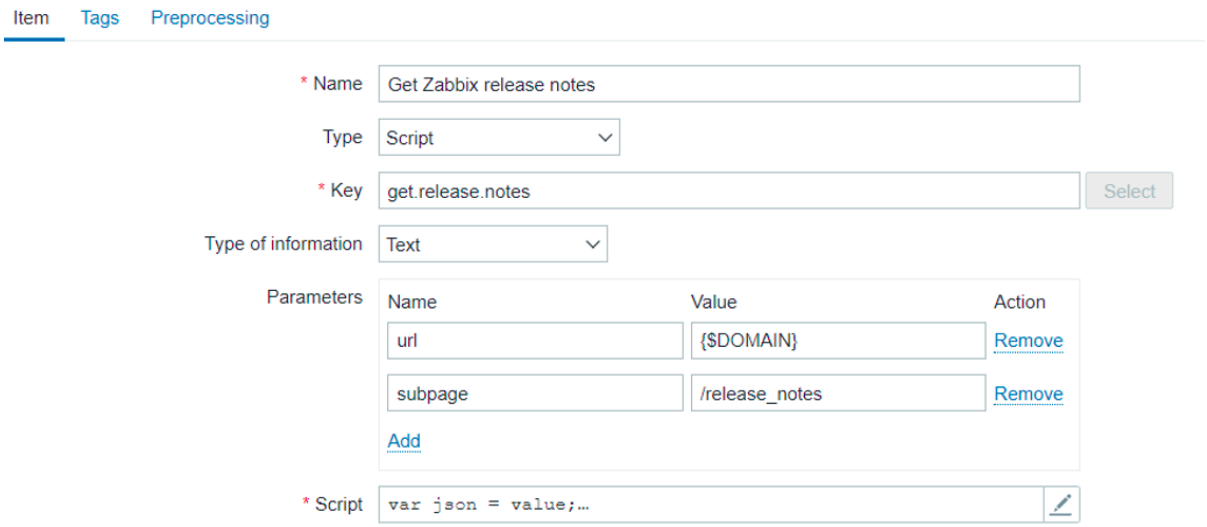
- В поле *Скрипт* введите следующий код:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Сбор данных с параметрами

Получение содержимого указанной страницы с использованием параметров:

- Создайте элемент данных с типом «Скрипт» и двумя параметрами:
- **url : {$DOMAIN}** (пользовательский макрос {$DOMAIN} должен быть определён, желательно на уровне узла сети)
- **subpage : /release_notes**



- В поле *Скрипт* введите следующий код:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Несколько HTTP запросов

Сбор содержимого как *https://www.example.com*, так и *https://www.example.com/release_notes*:

- Создайте элемент данных с типом «Скрипт».

- В поле *Скрипт* введите следующий код:

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Журналирование

Добавление записи «Log test» в журнал Zabbix сервера и получение при возврате «1» в качестве значения элемента данных:

- Создайте элемент данных с типом «Скрипт».

- В поле *Скрипт* введите следующий код:

```
Zabbix.log(3, 'Log test');
return 1;
```

Элемент данных браузера

Обзор

Элементы данных браузера позволяют отслеживать сложные веб-сайты и веб-приложения с помощью браузера.

Attention:
Поддержка элементов данных браузера в настоящее время является экспериментальной.

Элементы данных браузера собирают данные, выполняя заданный пользователем код JavaScript и получая данные по HTTP/HTTPS. Этот элемент данных может имитировать такие действия браузера, как щелчки, ввод текста, переходы по веб-страницам и другие действия пользователя на веб-сайтах или в веб-приложениях.

Помимо сценария, можно указать необязательный список параметров (пары имя и значение) и тайм-аут.

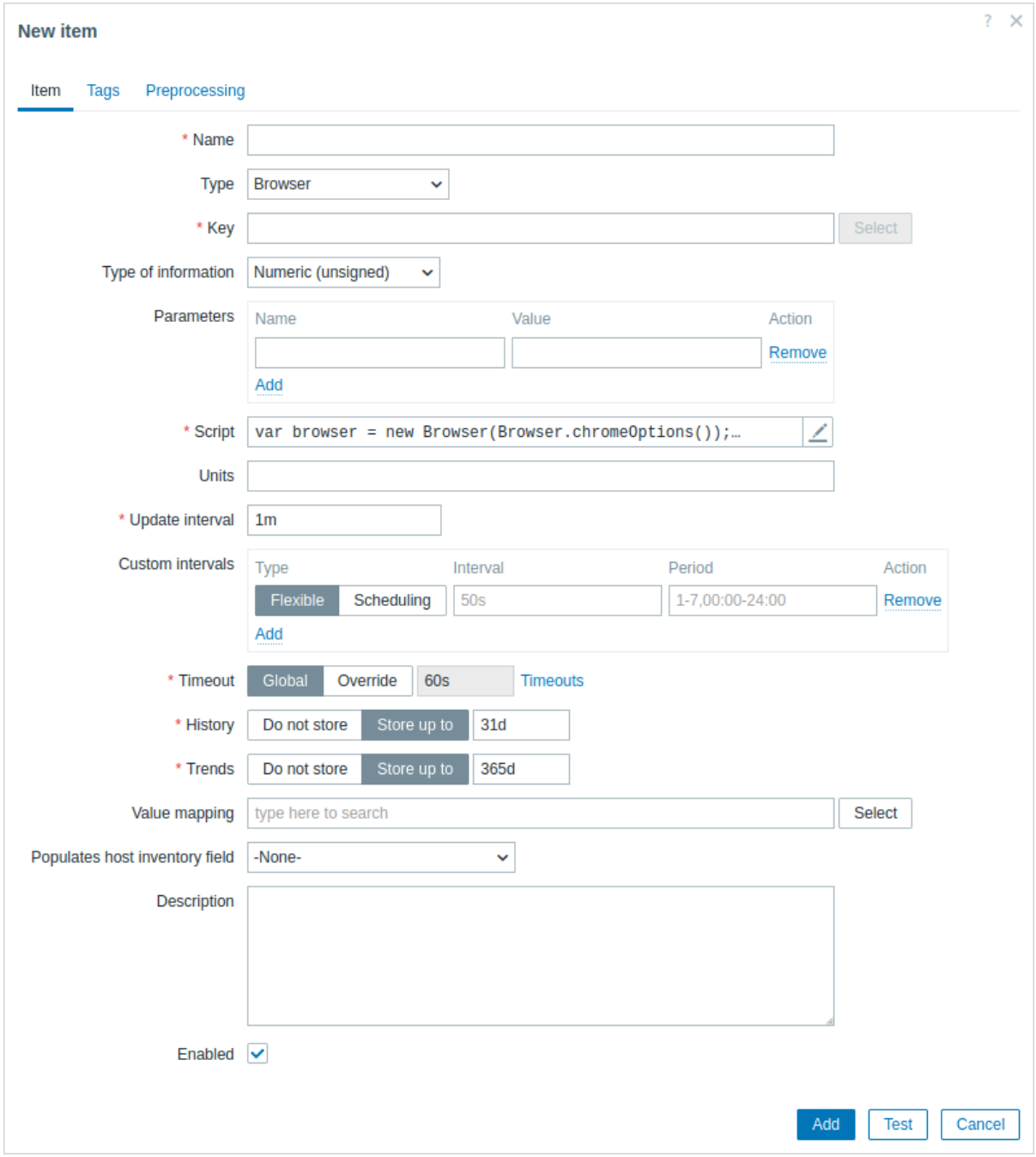
Attention:
Элемент данных частично реализует стандарт W3C WebDriver с использованием либо Selenium Server, либо обычного WebDriver (например, ChromeDriver) в качестве конечной точки веб-тестирования. Чтобы элемент данных работал, задайте параметр `WebDriverURL` в файле конфигурации Zabbix сервер/прокси (если используется ChromeDriver, см. Security Considerations). Для повышения производительности рекомендуется использовать выделенный сервер для среды веб-тестирования.

Проверки элементов данных браузера выполняются и обрабатываются браузерными опросчиками сервера или прокси Zabbix. При необходимости можно изменить количество предварительно созданных экземпляров браузерных опросчиков в параметре `StartBrowserPollers` файла конфигурации Zabbix сервер/прокси.

Для мониторинга сложных веб-сайтов и веб-приложений доступен шаблон Website by Browser в качестве готового к использованию шаблона.

Настройка

В поле *Тип (Type)* диалога настроек элемента данных выберите Браузер и заполните обязательные поля.



Все обязательные поля ввода отмечены красной звёздочкой.

Поля, требующие информации, специфичной для браузерного элемента данных:

Поле	Описание
Ключ (Key)	Введите уникальный ключ, который будет использоваться для идентификации элемента данных.
Параметры (Parameters)	Укажите переменные, которые будут переданы в скрипт как пары атрибут и значение. Поддерживаются пользовательские макросы. Чтобы узнать, какие встроенные макросы поддерживаются, выполните поиск по запросу «Браузерный элемент данных» в таблице поддерживаемых макросов.
Скрипт (Script)	Введите код JavaScript в модальном редакторе, который открывается при щелчке по полю параметра или по значку карандаша рядом с ним. Этот код должен предоставлять логику для возврата значения метрики. Код имеет доступ ко всем параметрам, всем дополнительным объектам JavaScript и объектам JavaScript браузерного элемента данных, добавленным Zabbix. Смотрите также: Руководство по JavaScript.

Поле	Описание
Время ожидания (Timeout)	Тайм-аут выполнения JavaScript (1-600s; при его превышении будет возвращена ошибка). Обратите внимание, что в зависимости от скрипта для срабатывания тайм-аута может потребоваться больше времени. Дополнительную информацию о параметре *Timeout* смотрите в разделе общие атрибуты элементов данных.

Примеры

Пример настройки Zabbix для мониторинга веб-сайтов с использованием шаблона *Website by Browser* см. в разделе Мониторинг веб-сайтов с помощью элементов данных Browser.

Скрипт по умолчанию

Следующий скрипт:

1. Инициализирует сессию браузера.
2. Переходит по указанному URL-адресу.
3. Собирает записи производительности и статистику сессии и возвращает их в виде строки JSON.

В поле *Скрипт (Script)* введите:

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Инициализация браузера с пользовательскими возможностями

Следующий скрипт:

1. Инициализирует сеанс браузера для доступного браузера на основе первого совпадающего браузера в порядке, указанном в скрипте.
2. Определяет возможности браузера, включая стратегию загрузки страницы и параметры, специфичные для каждого браузера, такие как режим headless для браузеров Chrome, Firefox и Microsoft Edge.

В поле *Script* введите:

```
var browser = new Browser({
    "capabilities":{
        "firstMatch":[
            {
                "browserName":"chrome",
                "pageLoadStrategy":"normal",
                "goog:chromeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"firefox",
                "pageLoadStrategy":"normal",
                "moz:firefoxOptions":{
                    "args":[
                        "--headless"
                    ]
                }
            },
            {
                "browserName":"MicrosoftEdge",
                "pageLoadStrategy":"normal",
```

```
                "ms:edgeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"safari",
                "pageLoadStrategy":"normal"
            }
        ]
    }
});
```

Инициализация браузера с GUI

По умолчанию сеансы браузера (за исключением Safari) инициализируются в режиме headless, то есть графический пользовательский интерфейс (GUI) браузера не отображается.

Следующий скрипт инициализирует сеанс браузера с включенным GUI.

Обратите внимание: если WebDriver не может найти исполняемый файл браузера, вы можете указать путь вручную.

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

//                  Firefox   GUI,                    :
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

//                  Microsoft Edge   GUI,                     :
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

> **Note:**
> Если ваши тесты выполняются на удаленном сервере или в контейнере, вы можете использовать клиент Virtual Network Computing (VNC), чтобы подключиться к VNC-серверу машины. Это позволяет удаленно просматривать графический интерфейс браузера и взаимодействовать с ним.

Создание снимков экрана

Следующий скрипт:

1. Инициализирует сеанс браузера.
2. Устанавливает размер области просмотра браузера, чтобы определить размер снимка экрана (указывается в параметрах, см. ниже).
3. Переходит по URL (указывается в параметре, см. ниже).
4. Собирает статистику сеанса, делает снимок экрана и добавляет его к собранной статистике.
5. Обрабатывает ошибки, фиксируя сообщения об ошибках и снимок экрана.
6. Возвращает собранные результаты в виде строки JSON.

Скрипт также использует параметры из формы настройки элемента данных:

- webURL - http://example.com
- width - 1920
- height - 1080

В поле *Script* введите:

```
var browser, result;

var browser = new Browser(Browser.chromeOptions());
```

```
try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.setScreenSize(Number(params.width), Number(params.height))

    browser.navigate(params.webURL);

    result = browser.getResult();
    result.screenshot = browser.getScreenshot();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Проверка входа в Zabbix

Следующий скрипт:

1. Инициализирует сессию браузера.
2. Переходит на страницу (указывается как параметр, смотрите ниже).
3. Вводит имя пользователя и пароль (указываются как параметры, смотрите ниже).
4. Находит и нажимает кнопку входа.
5. Находит и нажимает кнопку выхода.
6. Собирает данные о производительности до и после входа, а также после выхода.
7. Обрабатывает ошибки, фиксируя сообщения об ошибках и снимок экрана.
8. Возвращает собранные результаты в виде строки JSON.

Скрипт также использует параметры из диалога настроек элемента данных:

- webURL — http://{HOST.CONN}/index.php
- username — {$USERNAME}
- password — {$PASSWORD}

В поле *Скрипт (Script)* введите:

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); //            JSON,             ,          Zabbix.

    browser.navigate(params.webURL);
    browser.collectPerfEntries("open page");

    var el = browser.findElement("xpath", "//input[@id='name']");
    if (el === null) {
        throw Error("cannot find name input field");
    }
    el.sendKeys(params.username);

    el = browser.findElement("xpath", "//input[@id='password']");
    if (el === null) {
        throw Error("cannot find password input field");
    }
    el.sendKeys(params.password);

    el = browser.findElement("xpath", "//button[@id='enter']");
    if (el === null) {
```

```
        throw Error("cannot find login button");
    }
    el.click();

    browser.collectPerfEntries("login");

    el = browser.findElement("link text", "Sign out");
    if (el === null) {
        throw Error("cannot find logout button");
    }
    el.click();

    browser.collectPerfEntries("logout");

    result = browser.getResult();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Поиск ссылок

Следующий скрипт:

1. Инициализирует сеанс браузера.
2. Определяет функцию для удаления дублирующихся элементов из массива (см. шаг 5).
3. Переходит на страницу (указанную в параметрах, см. ниже).
4. Находит ссылки на странице.
5. Удаляет дублирующиеся ссылки, чтобы обеспечить их уникальность.
6. Извлекает только ссылки, начинающиеся с "http".
7. Форматирует извлеченные ссылки в определенную структуру.
8. Обрабатывает ошибки, захватывая сообщения об ошибках и снимок экрана.
9. Возвращает собранные результаты в виде строки JSON.

Скрипт также использует параметры из формы настройки элемента данных:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

В поле *Script* введите:

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value);  // Parse the JSON string containing parameters passed from Zabbix.

    function uniq(a) {
        return a.sort().filter(function (item, pos, ary) {
            return !pos || item != ary[pos - 1];
        });
    }

    browser.navigate(params.scheme + '://' + params.domain + params.path);

    var el = browser.findElements("link text", "");
    var links = [];
```

```
    for (var n = 0; n < el.length; n++) {
        links.push(el[n].getAttribute('href'));
    }

    links = uniq(links);

    result = [];
    for (i = 0; i < links.length; i++) {
        if (links[i].match(/^http.*/)) {
            var row = {};
            row["{#URL}"] = links[i];
            result.push(row);
        }
    }
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Запрос телеметрии

Этот тип элемента данных позволяет получать или агрегировать данные телеметрии. Данные телеметрии используются для мониторинга производительности приложений (APM), который помогает отслеживать прохождение запросов через приложения и службы, измерять их производительность и находить причины медленных или неудачных операций.

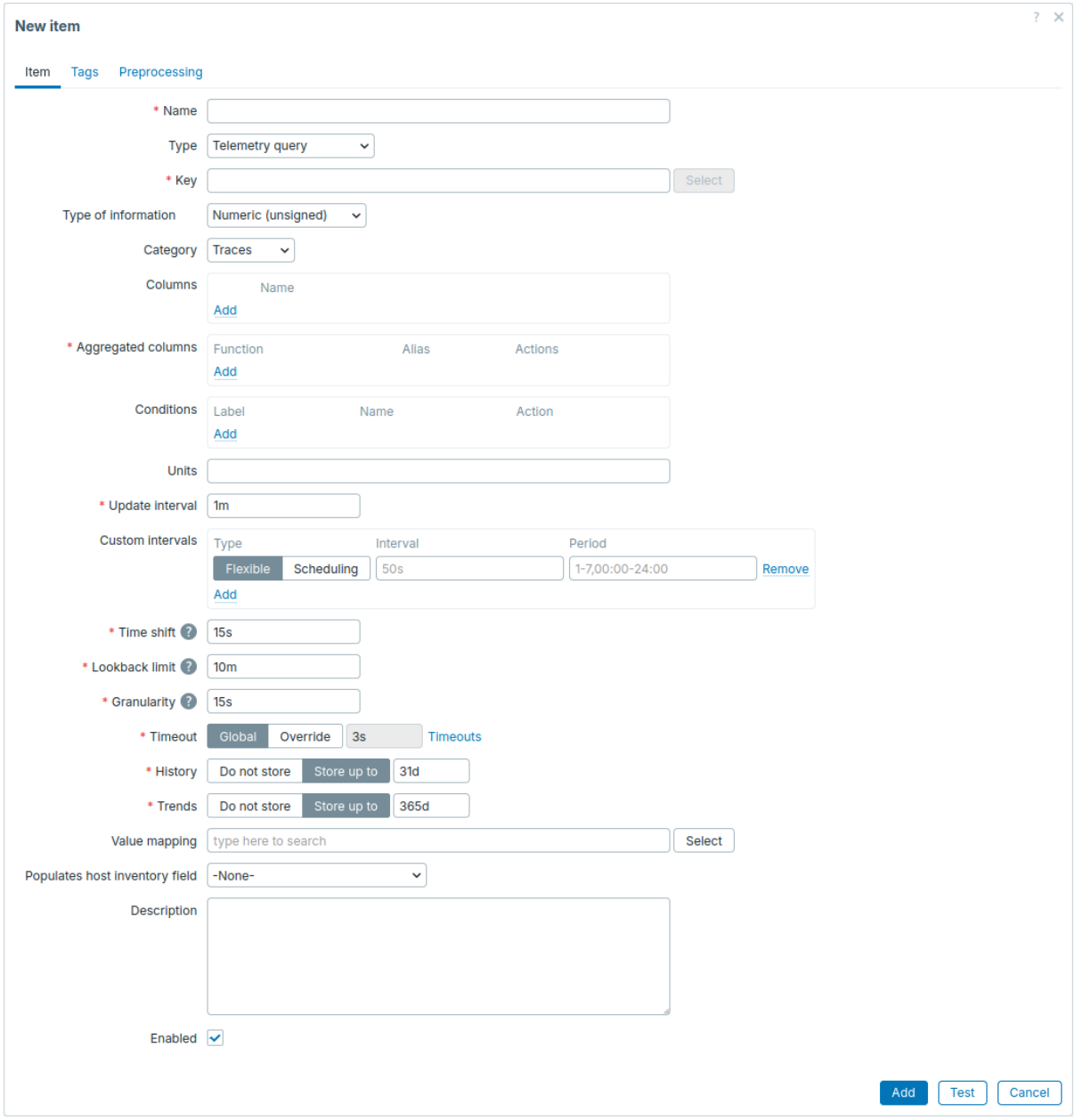
Данные телеметрии бывают трех типов: трассировки (путь, который запрос проходит через приложения и службы), метрики (измерения, например время ответа) и журналы (текстовые записи событий). С помощью элемента данных типа «Запрос телеметрии» можно, например, найти все трассировки, в которых служба вернула ошибку, получить среднее время ответа по метрикам или найти в журналах определенное сообщение об ошибке.

При настройке элемента данных типа «Запрос телеметрии» вы задаете набор инструкций (запрос), который указывает Zabbix, какие данные телеметрии необходимо запросить, как их отфильтровать и агрегировать. При выполнении проверки элемента данных Zabbix отправляет этот запрос источнику данных телеметрии и получает результат на его основе.

Чтобы использовать этот элемент данных, укажите источник данных телеметрии и задайте параметр `StartTelemetryQueryPoller` в конфигурации сервера Zabbix или прокси Zabbix в зависимости от того, какой из них отслеживает узел сети элемента данных.

Настройка

При создании элемента данных установите для него тип *Telemetry query*:



Параметр	Описание
Name	См. параметры, общие для всех элементов данных.
Type	Выберите *Telemetry query*.
Key	См. параметры, общие для всех элементов данных.
Type of information	
Category	Выберите тип сигнала, который требуется запросить: *Traces*, *Metrics* или *Logs*.
Metric points	Выберите тип точки метрики, который требуется запросить: *Sum*, *Gauge*, *Histogram* или *Exponential histogram*. Этот параметр доступен, если для *Category* установлено значение *Metrics*.
Columns	Добавьте столбцы данных телеметрии, которые требуется запросить. Для столбцов, содержащих пары «ключ-значение», также укажите имя ключа. Записи из этих столбцов будут включены в значение элемента данных без изменений.

Параметр	Описание
Aggregated columns	Добавьте столбцы данных телеметрии, которые требуется запросить, и укажите способ вычисления значения на основе их записей: **Function** — выберите функцию, применяемую к записям столбца: *min*, *max*, *avg*, *count*, *sum* или *percentile*. **Percentage** — укажите процентиль, который требуется вычислить на основе записей столбца (если для *Function* установлено значение *percentile*). **Column** — выберите столбец, на основе которого требуется выполнить вычисление (если для *Function* установлено значение, отличное от *count*). **Alias** — укажите уникальное имя вычисленного значения, используемое для его идентификации в значении элемента данных.
Type of calculation	Выберите способ оценки условий, заданных в параметре *Conditions* ниже: **And/Or** — должны выполняться все условия; условия с одинаковыми значениями *Type* и *Attribute key* объединяются оператором *Or*. **And** — должны выполняться все условия. **Or** — должно выполняться хотя бы одно условие. **Custom expression** — условия должны выполняться в соответствии с формулой, например `A and (B or C)`. Этот параметр доступен, если задано два или более условия.
Conditions	Добавьте условия для фильтрации записей столбца данных телеметрии: **Type** — выберите столбец данных телеметрии, к которому будет применяться условие. **Attribute key** — укажите ключ, к которому будет применяться условие (если для *Type* выбран столбец с парами «ключ-значение»). **Operator** — выберите сравнение, которое будет использоваться (см. ниже) между значением записи и *Attribute value*. **Attribute value** — укажите значение, с которым будет сравниваться значение записи (если для *Operator* установлено значение, отличное от *Exists*). Если для *Type* выбран столбец с одиночными значениями, условие будет применяться к записи столбца; если столбец содержит массив значений, запись будет соответствовать условию, если хотя бы одно значение в массиве удовлетворяет условию. Если для *Type* выбран столбец, содержащий пары «ключ-значение», условие будет применяться к значению ключа, указанного в *Attribute key*; если столбец содержит массив пар «ключ-значение», запись будет соответствовать условию, если хотя бы одна пара с указанным *Attribute key* удовлетворяет условию. Для столбцов с одиночными значениями можно использовать следующие операторы: **Equals** — выбирает записи, равные *Attribute value* (с учетом регистра). **Does not equal** — выбирает записи, не равные *Attribute value* (с учетом регистра). **Contains** — выбирает записи, содержащие *Attribute value* (с учетом регистра). **Does not contain** — выбирает записи, не содержащие *Attribute value* (с учетом регистра). Для столбцов, содержащих пары «ключ-значение», можно использовать следующие операторы: **Equals** — выбирает записи, в которых существует *Attribute key* и его значение равно *Attribute value* (с учетом регистра). **Does not equal** — выбирает записи, в которых существует *Attribute key* и его значение не равно *Attribute value* (с учетом регистра). **Exists** — выбирает записи, в которых существует *Attribute key* (с учетом регистра).
Units *Update interval* *Custom intervals*	См. параметры, общие для всех элементов данных.

Параметр	Описание
Time shift	Укажите, насколько далеко в прошлое (до текущего момента) должна заканчиваться временная область, запрашиваемая элементом данных. Поддерживается число секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательские макросы. Каждый запрос получает данные за определенную временную область. Обычно область начинается с момента окончания временной области предыдущего запроса и заканчивается в точке, заданной параметром *Time shift*. Если временная область предыдущего запроса закончилась позже, чем текущий момент минус *Granularity*, временная область текущего запроса начинается с текущего момента минус *Granularity*, перекрываясь таким образом с временной областью предыдущего запроса. Если предыдущего запроса, с которого можно продолжить, нет, область начинается в точке, заданной параметром *Lookback limit*. Если этот узел сети начинает отслеживаться другой сущностью (например, другим прокси из группы прокси), элемент данных может больше не знать, когда завершился предыдущий запрос. Это может привести к появлению дублирующихся значений.
Lookback limit	Укажите, насколько далеко в прошлое (до текущего момента) элемент данных должен запрашивать данные, если предыдущего запроса, с которого можно продолжить, нет (например, при первом запуске элемента данных) или если предыдущий запрос был выполнен раньше, чем *Update interval* (например, из-за недоступности сервера Zabbix). Поддерживается число секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательские макросы. Если источник данных телеметрии содержит большой объем данных (например, более миллиона записей в час), большое значение *Lookback limit* может значительно снизить производительность при его использовании элементом данных.
Granularity	Укажите длительность фрагмента, на который разбивается полный временной диапазон запроса. Поддерживается число секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательские макросы.
Timeout *History* *Trends* *Value mapping* *Populates host inventory field* *Description* *Enabled*	См. параметры, общие для всех элементов данных.

Поддерживаемые столбцы данных телеметрии

В этом разделе перечислены столбцы, поддерживаемые в следующих параметрах элемента данных: *Columns*, *Aggregated columns* и *Conditions*. Это столбцы, которые ClickHouse использует для хранения данных телеметрии в соответствии со схемой экспортера ClickHouse OpenTelemetry.

Выберите *категорию*:

- Трассировки
- Метрики: сумма
- Метрики: измерение
- Метрики: гистограмма
- Метрики: экспоненциальная гистограмма
- Журналы

Трассировки

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
Длительность	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Метка времени	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Метрики: сумма

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Метрики: Gauge

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
Атрибуты	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Флаги	✓	–	–
Описание метрики	✓	–	✓
Имя метрики	✓	–	✓
Единица измерения метрики	✓	–	✓
Атрибуты ресурса	✓	–	✓
URL схемы ресурса	✓	–	✓
Атрибуты области действия	✓	–	✓
Количество отброшенных атрибутов области действия	✓	✓	–
Имя области действия	✓	–	✓
URL схемы области действия	✓	–	✓
Версия области действия	✓	–	✓
Имя службы	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Значение	✓	✓	–

Метрики: гистограмма

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Метрики: экспоненциальная гистограмма

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Журналы

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓

Столбец	*Столбцы*	*Агрегированные столбцы*	*Условия*
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

История и тренды

Обзор

История и динамика изменений — два способа хранения в Zabbix собранных данных.

В то время как история хранит каждое собранное значение, динамика изменений хранит усреднённую информацию за каждый час, что, следовательно, менее требовательно к ресурсам.

Хранение истории

Вы можете указать, как много дней будет храниться история:

- в диалоге свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач очистки истории

Любые более старые данные будут удалены с помощью автоматической очистки базы данных (Housekeeper).

Общий настоятельный совет — хранить историю как можно меньшее количество дней и, таким образом, не перегружать базу данных большим количеством значений истории.

Вместо долговременного хранения истории вы можете хранить больше данных динамики изменений. Например, вы можете хранить историю 14 дней, а динамику изменений — 5 лет.

Вы можете прикинуть, сколько потребуется места для истории в сравнении с динамикой изменений, используя страницу оценки размера базы данных.

При меньшем хранении истории, вы всё равно сможете просматривать старые данные на графиках, поскольку для отображения старых данных графики будут использовать значения динамики изменений.

Attention:
Если история установлена в значение «0», то элемент данных будет обновлять только зависимые элементы данных и поля инвентарных данных. Функции триггера не будут вычисляться, так как триггерные вычисления основаны только на данных истории.

Note:
В качестве альтернативного подхода к хранению истории ознакомьтесь с функционалом экспорта истории в подгружаемых модулях.

Хранение динамики изменений

Динамика изменений является встроенным механизмом обработки данных истории, где по числовым типам данных для каждого часа хранятся значения минимума, максимума, среднее и общее количество значений.

Вы можете указать, как много дней будет храниться динамика изменений:

- в диалоге свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач очистки истории

Обычно динамика изменений может храниться намного дольше чем история. Любые более старые данные будут удалены при помощи автоматической очистки базы данных (Housekeeper).

Во время работы, по мере поступления новых значений, Zabbix сервер собирает и хранит динамику изменений в специальном кэше динамики изменений. Сервер сбрасывает динамику изменения **предыдущего часа** каждого элемента данных в базу данных (при этом она становится доступной из веб-интерфейса) в следующих случаях:

- сервер получает первое за текущий час значение элемента данных
- до конца текущего часа остаётся 5 или менее минут и всё ещё не пришло ни одно значение элемента данных за текущий час
- сервер останавливается

Таким образом, самое раннее, когда можно увидеть новое значение динамики изменения текущего часа в веб-интерфейсе — это начало следующего часа (если элемент данных обновляется регулярно), а самое позднее — конец следующего часа (если элемент данных обновляется редко), что даёт 2 часа максимум.

Если в момент сбрасывания значения динамики изменения сервер замечает, что значение за соответствующий час в базе данных уже имеется (например, сервер был перезапущен в середине часа), ему, возможно, придётся выполнять SQL инструкции обновления вместо вставки. Поэтому на больших инсталляциях, если необходим перезапуск сервера, постарайтесь спланировать его так, чтобы остановка сервера произошла в конце часа, а запуск — в начале следующего, чтобы избежать перекрытие данных динамики изменений.

При работе со значениями динамики изменений Zabbix сервер к таблицам истории базы данных не обращается.

Attention:
Если динамика изменений установлена в значение «0», Zabbix сервер вообще не будет считать и хранить динамику изменений.

Note:
Динамика изменений вычисляется и сохраняется с использованием того же типа данных, что и исходные значения. В результате среднее значение вычисления значений целочисленных типов данных округляется, и чем меньше интервал значений, тем меньше будет точность результата. Например, если элемент данных имеет значения 0 и 1, то средним значением будет 0, а не 0.5.
Перезапуск сервера также может привести к потере точности вычислений средних значений у целочисленных типов данных за текущий час.

Пользовательские параметры

Overview

A user parameter lets you define a custom item key and the command used to retrieve its value when no built-in Zabbix agent item provides the data you need. A user parameter is a user-defined command referenced by a user-defined item key and executed by the Zabbix agent when the corresponding item is checked.

Note:
On UNIX systems, the agent executes user parameter commands using `/bin/sh`, under the operating system account running the agent (usually `zabbix`). That account must have sufficient permissions to execute the command and access any files, sockets, or credentials it uses.

There are two types of user parameters:

- **Simple user parameters** do not accept arguments; they define a fixed item key and a command to execute.
- **Flexible user parameters** accept arguments (specified as parameters in the item key) and pass them to the command, allowing a single definition to serve multiple items.

User parameter configuration

A user parameter is defined using the `UserParameter` configuration parameter.

A user parameter has the following syntax:

`UserParameter=<key>,<shell command>`

where:

- <key> - a unique item key that identifies the user parameter.
 When configuring an item, specify this key to reference the command defined by the user parameter.
 For a simple user parameter, the key is fixed and does not accept arguments. For a flexible user parameter, append `[*]` to the key (for example, `mysql.status[*]`) to indicate that it accepts arguments. Specify these arguments as parameters in the item key when configuring the item. The command can refer to the parameters using positional references $1–$9.

- <shell command> - a shell command or a path to a custom script that the agent executes to obtain the item value.
 For the flexible user parameter, the command can use positional references $1-$9 to refer to the corresponding parameters in the item key (for example, UserParameter=mysql.status[*],mysqladmin status --user=$1 --password=$2). Zabbix parses the parameters enclosed in [] in the item key (i.e., mysql.status[zabbix,{$MYSQL.PASSWORD}]) and substitutes the positional references in the command accordingly.
 $0 is substituted with the original command before positional references are expanded. Positional references are interpreted regardless of whether they are enclosed in double (") or single (') quotes. To use positional references without substitution, specify a double dollar sign. For example, awk '{print $$2}' passes $2 to the command unchanged.
 Note that positional references prefixed with $ are searched for and replaced by the Zabbix agent only in flexible user parameters. For simple user parameters, positional references are not processed, so escaping $ is unnecessary.

Attention:
Certain characters cannot be passed as arguments by default. For the complete list of characters, see UnsafeUserParameters.

UserParameter must be configured on each Zabbix agent that uses it: - Directly in the Zabbix agent configuration file (zabbix_agentd.conf or zabbix_agent2.conf). - In a separate *.conf file, provided that the file is included by the agent configuration.

More than one UserParameter can be defined per agent, as multiple files can be included with different sets of parameters. Each user parameter must have a unique item key within the monitored host.

The UserParameterDir configuration parameter lets you specify the working directory for executing UserParameter commands. If configured, the agent changes its working directory to the specified directory before executing a command. Commands can be specified using a relative path with the ./ prefix, for example, UserParameter=mykey,./myscript.sh.

To configure a user parameter:

1. Write a custom script or command to retrieve the required data.
2. Add the command and its unique item key to the UserParameter configuration parameter in the Zabbix agent configuration file (zabbix_agentd.conf).
3. Reload user parameters from the configuration file by running the runtime control command:

   zabbix_agentd -R userparameter_reload

   Alternatively, restart the agent to apply the configuration changes.
4. Create a new item on the monitored host with the item key defined in UserParameter.
 Set the item's type to *Zabbix Agent* or *Zabbix Agent (active)*.
 Make sure that the item's *Type of information* is configured correctly; otherwise, Zabbix will reject the returned value.

See Extending Zabbix agents for an example of configuring and using user parameters.

User parameters are subject to the Zabbix agent check timeout. If the timeout is reached, the process executing the user parameter is terminated. See Command execution for more information.

Note that long-running scripts may consume system resources and delay other checks.

Результат команды

Возвращаемое значение команды — это стандартный вывод вместе со стандартным потоком ошибок, созданным командой.

Attention:
Элемент данных, который возвращает текст (символьный, логовый или текстовый тип информации), не станет неподдерживаемым в случае вывода в стандартный поток ошибок.

Возвращаемое значение ограничено 16 МБ (включая завершающие пробелы, которые обрезаются); ограничения базы данных также применяются.

Пользовательские параметры, которые возвращают текст (символьный, логовый или текстовый тип информации), также могут возвращать пробелы. В случае недопустимого результата элемент данных станет неподдерживаемым.

Примеры простых пользовательских параметров

Простая команда:

```
UserParameter=ping,echo 1
```

Для элемента данных с ключом «ping» агент будет всегда возвращать «1».

Более сложный пример:

```
UserParameter=mysql.ping,mysqladmin -uroot ping|grep -c alive
```

Агент будет возвращать «1», если MySQL сервер доступен, «0» — в противном случае.

Пример 1

Очень простой пример:

```
UserParameter=ping[*],echo $1
```

Мы можем создать неограниченное количество элементов данных для наблюдения за чем угодно, указав в таком формате: ping[что угодно].

- ping[0] — будет всегда возвращать «0»
- ping[aaa] — будет всегда возвращать «aaa»

Пример 2

Давайте добавим больше смысла!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

Этот параметр можно использовать для мониторинга доступности баз данных MySQL. Параметрами мы можем передать имя пользователя и пароль:

```
mysql.ping[zabbix,our_password]
```

Пример 3

Сколько строк в файле, которые соответствуют регулярному выражению?

```
UserParameter=wc[*],grep -c "$2" $1
```

Этот параметр можно использовать для подсчёта количества строк в файле.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Расширение возможностей агентов Zabbix

В этом руководстве приведены пошаговые инструкции по расширению функциональности агента Zabbix с помощью пользовательского параметра.

Шаг 1

Напишите скрипт или команду для получения требуемого параметра.

Например, можно написать следующую команду, чтобы получить общее количество запросов, выполненных сервером MySQL:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

При выполнении команда возвращает общее количество SQL-запросов.

Шаг 2

Добавьте команду в файл конфигурации агента Zabbix (`zabbix_agentd.conf`):

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions — это уникальный идентификатор. Он может быть любым допустимым идентификатором ключа, например *queries*.

Проверьте этот параметр с помощью агента Zabbix, используя флаг `-t` (однако при запуске от имени пользователя root обратите внимание, что у агента могут быть другие разрешения при запуске в качестве демона):

```
zabbix_agentd -t mysql.questions
```

Шаг 3

Перезагрузите пользовательские параметры из файла конфигурации, выполнив:

```
zabbix_agentd -R userparameter_reload
```

Вместо команды управления во время работы можно также перезапустить агент.

Проверьте параметр с помощью утилиты zabbix_get.

Шаг 4

Добавьте новый элемент данных с Key=mysql.questions к контролируемому узлу сети. Тип элемента данных должен быть либо *Zabbix Agent*, либо *Zabbix Agent (active)*.

Имейте в виду, что тип возвращаемых значений должен быть правильно задан на сервере Zabbix. В противном случае Zabbix не примет их.

Счетчики производительности Windows

Обзор

Вы можете эффективно отслеживать счетчики производительности Windows с помощью ключа perf_counter[].

Например:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

или

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Дополнительную информацию об использовании этого ключа или его эквивалента только на английском языке perf_counter_en см. в разделе Ключи элементов данных, специфичные для Windows.

Чтобы получить полный список счетчиков производительности, доступных для мониторинга, можно выполнить команду:

```
typeperf -qx
```

Также можно использовать низкоуровневое обнаружение для обнаружения нескольких экземпляров объектов счетчиков производительности Windows и автоматизировать создание элементов данных perf_counter для объектов с несколькими экземплярами.

Числовое представление

Windows поддерживает числовые представления (индексы) имен объектов и счетчиков производительности. Zabbix поддерживает эти числовые представления в качестве параметров ключей элементов данных perf_counter, perf_counter_en, а также параметров конфигурации PerfCounter, PerfCounterEn.

Однако не рекомендуется использовать их, если вы не можете гарантировать, что ваши числовые индексы соответствуют правильным строкам на определенных узлах сети. Если вам необходимо создать переносимые элементы данных, работающие на разных узлах сети с различными локализованными версиями Windows, можно использовать ключ perf_counter_en или параметр конфигурации PerfCounterEn, которые позволяют использовать английские имена независимо от локали системы.

Чтобы узнать числовые эквиваленты, запустите regedit, затем найдите *Counter* в HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wi
NT\CurrentVersion\Perflib\009.

Запись реестра содержит такие данные:

```
1
1847
2
System
4
Memory
6
% Processor Time
10
File Read Operations/sec
12
File Write Operations/sec
14
File Control Operations/sec
16
File Read Bytes/sec
18
File Write Bytes/sec
```

....

Здесь можно найти соответствующие номера для каждой строковой части счетчика производительности, например в `\System\% Processor Time`:

```
System → 2
% Processor Time → 6
```

Затем эти номера можно использовать для представления пути в числовом виде:

```
\2\6
```

Параметры счетчиков производительности

Для мониторинга счетчиков производительности Windows можно развернуть некоторые параметры PerfCounter.

Например, их можно добавить в файл конфигурации агента Zabbix:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
or
PerfCounter=UserPerfCounter2,"\4\24",30
```

После добавления таких параметров можно просто использовать `UserPerfCounter1` или `UserPerfCounter2` в качестве ключей при создании соответствующих элементов данных.

Не забудьте перезапустить агент Zabbix после внесения изменений в файл конфигурации.

Массовое обновление

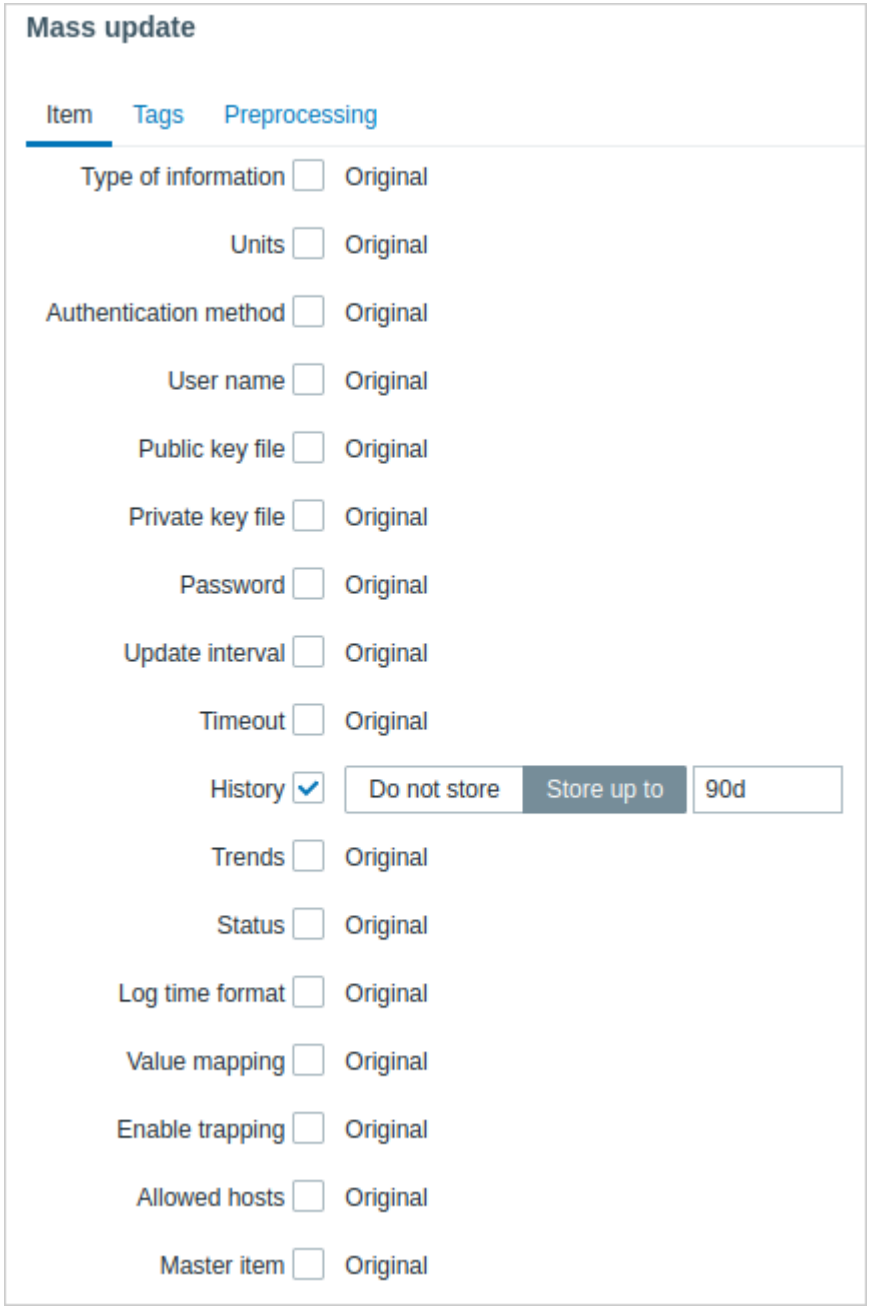
Обзор

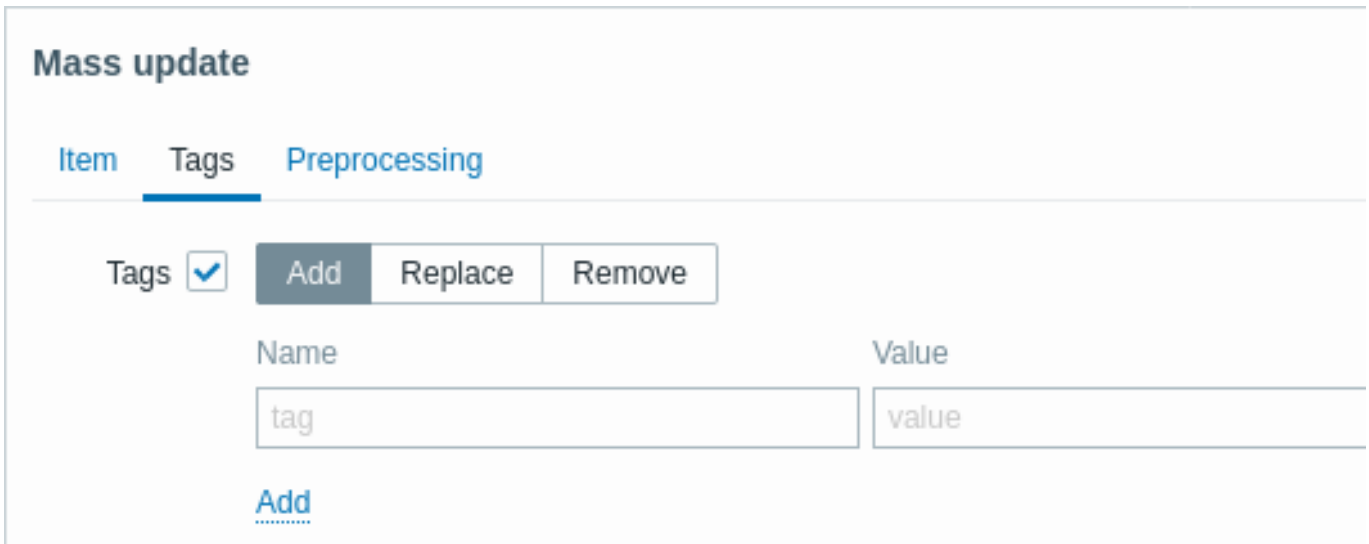
Иногда вы можете захотеть изменить какие-нибудь атрибуты у нескольких элементов данных одновременно. Вместо того, чтобы открывать на редактирование каждый элемент данных по отдельности, вы можете воспользоваться для этого функцией массового обновления.

Массовое обновление

Чтобы выполнить массовое обновление элементов данных, выполните следующие действия:

- Установите флажки элементов данных, которые необходимо обновить, в списке
- Нажмите *Массовое обновление* под списком
- Перейдите на вкладку с необходимыми атрибутами (*Элемент данных*, *Теги* или *Предобработка*)
- Установите флажки атрибутов, которые необходимо обновить
- Введите новые значения атрибутов



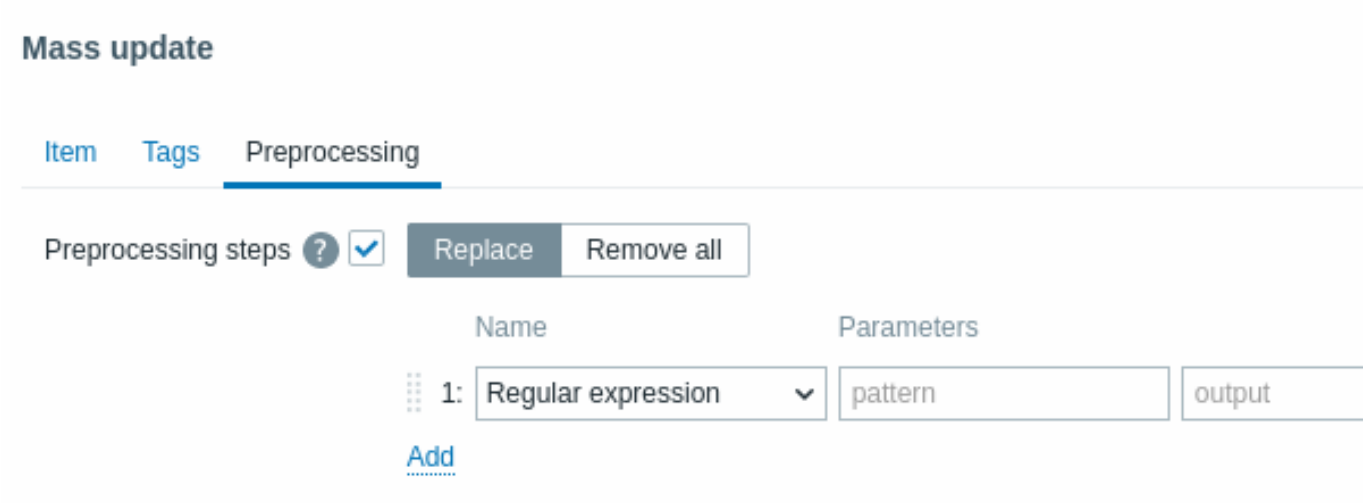


Опция *Теги* позволяет:

- *Добавить* - добавить указанные теги к элементам данных (теги с одинаковым именем, но разными значениями не считаются «дубликатами» и могут быть добавлены к одному элементу данных).
- *Заменить* - удалить указанные теги и добавить теги с новыми значениями

- *Удалить* - удалить указанные теги из элементов данных

В тегах поддерживаются пользовательские макросы, макросы {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}.



Установите флажок рядом с пунктом *Шаги предобработки*, чтобы активировать массовое обновление шагов предобработки.

Массовое обновление *Предобработки* позволяет:

- *Заменить* - заменить существующие шаги предобработки элементов данных шагами предобработки, указанными ниже
- *Удалить все* - удалить все шаги предобработки из элементов данных

После завершения нажмите *Обновить*.

Сопоставление значений

Обзор

Для более «человеческого» представления полученных значений, вы можете использовать преобразования значений, которые содержат соответствия между числовыми либо строковыми значениями и строковыми представлениями.

Например, элемент данных, который имеет значения «0» или «1», может использовать преобразование значений для представления этих значений в более читабельном виде:

- 0 → Недоступен
- 1 → Доступен

Преобразование значений для видов резервного копирования данных может быть настроено следущим образом:

- F → Полный
- D → Дифференциальный
- I → Инкрементный

Диапазоны значений напряжения можно отображать так:

- <=209 → Низкое
- 210-230 → OK
- >=231 → Высокое

Преобразования значений используются в веб-интерфейсе Zabbix и в оповещениях, отсылаемых по каналам оповещений.

Attention:
Подстановка полученного значения его настроенным представлением выполняется как веб-интерфейсом Zabbix, так и сервером; однако, сервер обрабатывает подстановки только в следующих случаях:

- при заполнении инвентарных данных узла сети;
- при раскрытии поддерживаемых макросов {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} и {EVENT.CAUSE.OPDATA}.

Преобразования значений настраиваются на уровне шаблонов или узлов сети. Сразу после добавления, эти преобразования значений становятся доступными для всех элементов данных соответствующего шаблона или узла сети. При

настройке элементов данных выберите в параметре *Преобразование значений (Value mapping)* имя ранее добавленного преобразования значений.

Note:
Наследование преобразований значений отсутствует — узлы сети и шаблоны не наследуют преобразования значений от присоединённых шаблонов. Элементы данных шаблона на узле сети продолжат использовать преобразования значений, настроенные в своём шаблоне.

Note:
Преобразования значений можно использовать с элементами данных, которые имеют типы информации *Числовой (целое положительное)*, *Числовой (с плавающей точкой)* и *Символ*.

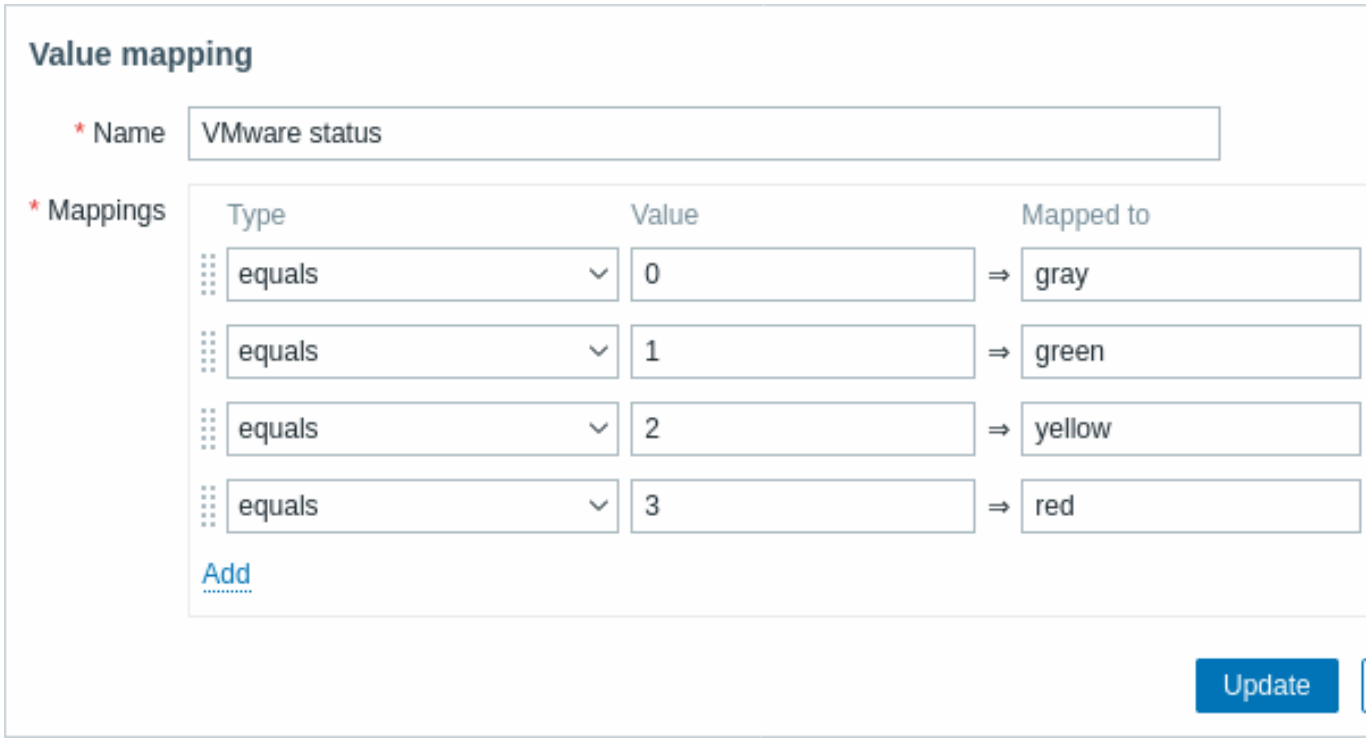
Преобразования значений экспортируются/импортируются вместе с соответствующим шаблоном или узлом сети. Их можно обновлять массово, используя диалоги массового обновления узла сети и шаблона.

Настройка

Чтобы настроить сопоставление значений, выполните следующие действия:

1. Откройте форму настройки узла сети или шаблона.

2. На вкладке *Сопоставление значений* нажмите *Добавить*, чтобы добавить новое сопоставление значений, или нажмите имя существующего сопоставления, чтобы изменить его.



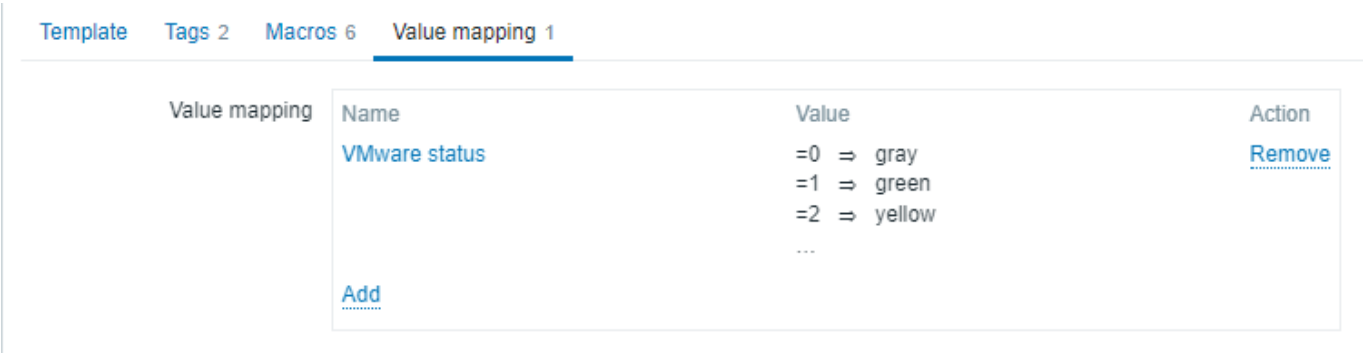
Параметры сопоставления значений:

Параметр	Описание
Имя	Уникальное имя набора сопоставлений значений.
Сопоставления	Отдельные правила сопоставления числовых значений или строковых значений со строковыми представлениями. Сопоставление выполняется в порядке следования правил, который можно изменить путем перетаскивания.

Параметр	Описание
Тип	Тип сопоставления: **равно** - будут сопоставлены равные значения; **больше или равно** - будут сопоставлены равные или большие значения; **меньше или равно** - будут сопоставлены равные или меньшие значения; **в диапазоне** - будут сопоставлены значения в диапазоне; диапазон указывается в формате <number1>-<number2> или <number>; поддерживается несколько диапазонов (например, 1-10,101-110,201); **регулярное выражение** - будут сопоставлены значения, соответствующие регулярному выражению (глобальные регулярные выражения не поддерживаются); **по умолчанию** - будут сопоставлены все остальные значения, кроме значений с определенными сопоставлениями. Для сопоставления диапазонов поддерживаются только числовые типы значений (*больше или равно*, *меньше или равно*, *в диапазоне*).
Значение	Входящее значение (может содержать диапазон или регулярное выражение в зависимости от типа сопоставления).
Сопоставляется с	Строковое представление (до 64 символов) входящего значения.

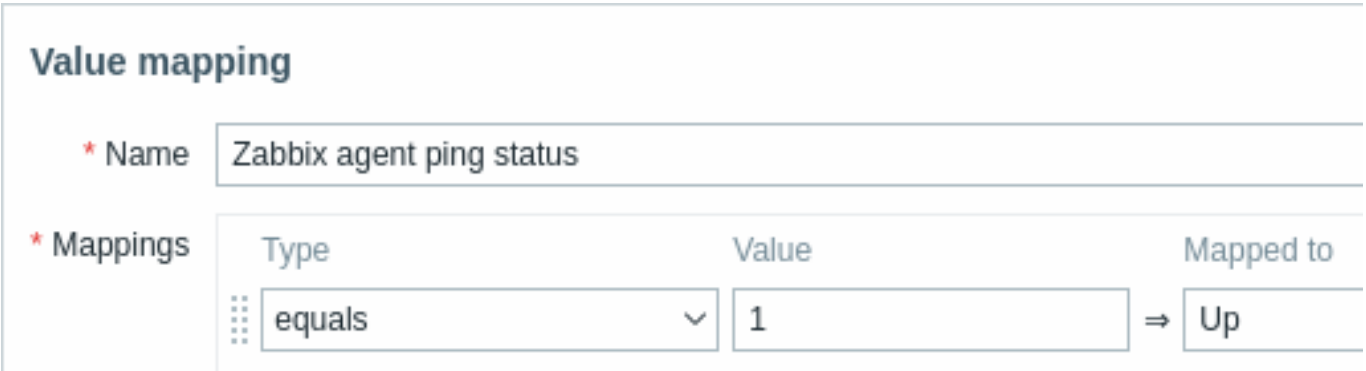
Все обязательные поля ввода отмечены красной звездочкой.

При просмотре сопоставления значений в списке отображаются только первые три сопоставления; три точки указывают на наличие дополнительных сопоставлений.



Пример сопоставления значений

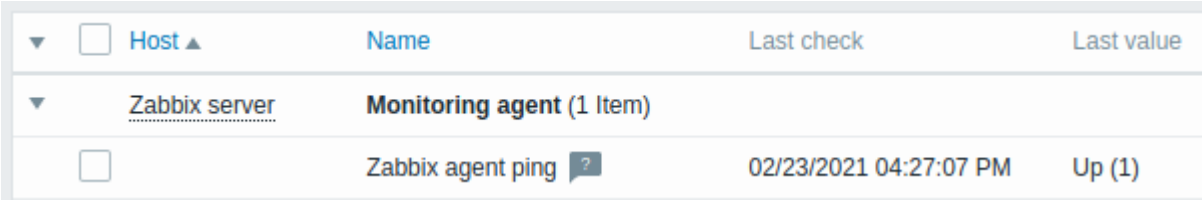
Один из предопределенных элементов данных агента *Zabbix agent ping* использует сопоставление значений на уровне шаблона "Zabbix agent ping status" для отображения своих значений.



В форме настройки элемента данных ссылку на это сопоставление значений можно найти в поле *Сопоставление значений*:



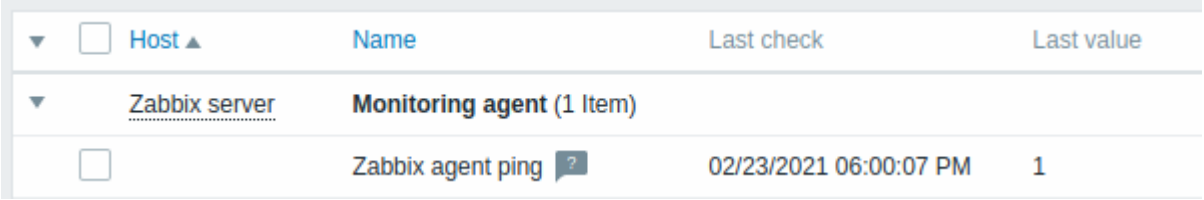
Это сопоставление используется в разделе *Мониторинг* > *Последние данные* для отображения значения "Up" (исходное значение отображается в скобках).



Host ▲	Name	Last check	Last value
Zabbix server	**Monitoring agent** (1 Item)		
	Zabbix agent ping	02/23/2021 04:27:07 PM	Up (1)

Note:
В разделе *Последние данные* отображаемые значения сокращаются до 20 символов. Если используется сопоставление значений, сокращение применяется не к сопоставленному значению, а только к исходному значению (отображаемому в скобках).

Без предопределенного сопоставления значений отображалось бы только значение ”1”, которое может быть трудно интерпретировать.



Host ▲	Name	Last check	Last value
Zabbix server	**Monitoring agent** (1 Item)		
	Zabbix agent ping	02/23/2021 06:00:07 PM	1

Очередь

Обзор

Очередь отображает элементы данных, ожидающих обновления. Очередь — просто **логическое** представление данных. Это не очередь IPC и не какой-то другой механизм очереди в Zabbix.

Элементы данных, наблюдаемые через прокси, также включены в очередь — они будут считаться задерживающимися за период обновления данных истории с прокси.

В очереди отображаются только те элементы данных, которые имеют запланированное время обновления. Это означает, что следующие типы элементов данных исключены из очереди:

- активные элементы данных Zabbix агента log, logrt и журнал событий
- элементы данных SNMP трапов
- траппер элементы данных
- элементы данных веб-мониторинга
- зависимые элементы данных

Статистика, представленная в очереди, является хорошим показателем производительности сервера Zabbix.

Очередь запрашивается напрямую с Zabbix сервера с использованием протокола JSON. Информация доступна, только если Zabbix сервер запущен.

Attention:
Элементы данных не учитываются в очереди, если интерфейс элемента данных становится недоступным из-за проблем с подключением или агент не работает должным образом.

Чтение очереди

Чтобы прочитать очередь, перейдите в раздел *Администрирование* > *Очередь* > *Обзор очереди*.

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

На этом изображении в целом отображается состояние «ok», поэтому можно предположить, что сервер работает нормально.

В очереди отображаются некоторые элементы данных, ожидающие обработки до 30 секунд. Было бы полезно узнать, что это за элементы данных.

Для этого выберите *Подробности очереди* в раскрывающемся списке заголовка. Теперь вы увидите список этих задержанных элементов данных.

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

Эти сведения могут помочь выяснить, почему обработка этих элементов данных задерживается.

Наличие одного или двух задержанных элементов данных, возможно, не является поводом для беспокойства. Они могут обновиться через секунду. Однако если вы видите множество элементов данных, обработка которых задерживается надолго, это может указывать на более серьезную проблему.

См. также: Синхронизация часовых поясов при использовании интервалов планирования.

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

Элемент данных очереди

Для мониторинга состояния очереди в Zabbix можно использовать специальный внутренний элемент данных `zabbix[queue,<from>`. Он возвращает количество элементов данных, задержанных на заданный промежуток времени. Дополнительную информацию см. в разделе Внутренние элементы данных.

Кэш значений

Обзор

Чтобы делать вычисления выражений триггеров, вычисляемых элементов данных и некоторых макросов намного быстрее, в Zabbix сервере поддерживается опция кэша значений.

Этот находящийся в оперативной памяти кэш используется для доступа к данным истории, вместо выполнения прямых SQL-запросов к базе данных. Если данные истории не присутствуют в кэше, недостающие значения запрашиваются из базы данных, а кэш обновляется соответственно.

Значения элементов данных остаются в кэше значений до тех пор, пока не произойдёт одно из следующих событий:

- элемент данных будет удалён (кэшированные значения удаляются после следующей синхронизации конфигурации);
- значение элемента данных выйдет за пределы диапазона времени или количества, указанного в выражении триггера/вычисляемого элемента данных (кэшированное значение удаляется при получении нового значения);
- диапазон времени или количества, указанный в выражении триггера/вычисляемого элемента данных, изменён таким образом, что для расчёта требуется меньше данных (ненужные кэшированные значения удаляются через 24 часа).

Note:
Состояние кэша значений можно наблюдать с помощью опции управления работой сервера `diaginfo` (или `diaginfo=valuecache`) и проверкой раздела кэша значений на предмет диагностической информации. Это может быть полезно для определения неправильно настроенных триггеров или вычисляемых элементов данных.

Для включения функции кэша значений можно указать необязательный параметр `ValueCacheSize` в конфигурации сервера Zabbix.

Для мониторинга кэша значений поддерживаются два внутренних элемента данных: `zabbix[vcache,buffer,<mode>]` и `zabbix[vcache,cache,<parameter>]`. Подробнее см. в разделе внутренние элементы данных.

Выполнить сейчас

Обзор

Опция *Выполнить сейчас (Execute now)* позволяет выполнить пассивную проверку немедленно.

Сбор значений элементов данных в Zabbix'е основывается на заданных интервалах обновления. Некоторые элементы данных (например, правила низкоуровневого обнаружения) имеют довольно продолжительные интервалы, а вам может понадобиться извлечь новое значение незамедлительно — например, чтобы заметить изменения в обнаруживаемых ресурсах.

Опция *Выполнить сейчас (Execute now)* поддерживается для следующих типов элементов данных:

- Zabbix агент (пассивный)
- Простая проверка
- SNMP агент (v1/v2/v3)
- Zabbix внутренний
- Внешняя проверка
- Монитор баз данных
- HTTP агент
- IPMI агент
- SSH агент
- TELNET агент
- JMX агент
- Вычисляемый
- Зависимый элемент данных
- Скрипт
- Браузер

Если элемент данных относится к типу «Зависимый элемент данных (Dependent item)», то его основной элемент данных также должен быть одного из вышеуказанных типов.

Конфигурация

Вы можете выполнять пассивные проверки из следующих разделов:

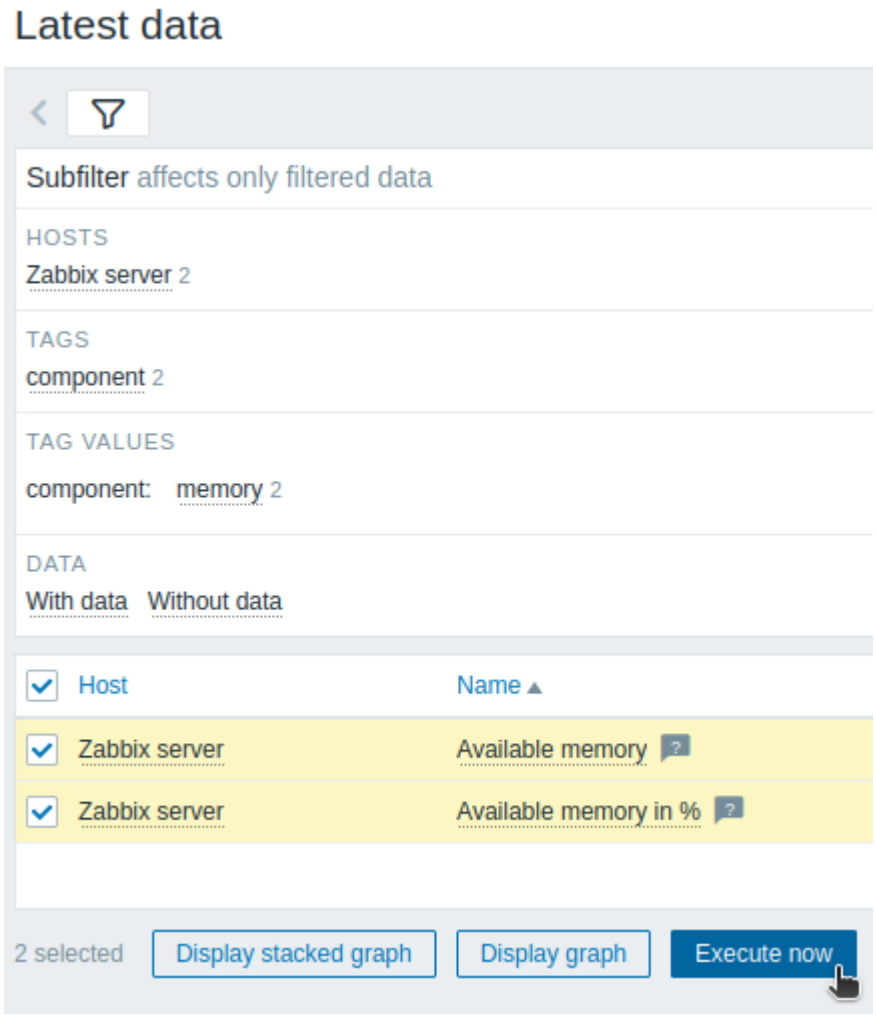
- *Мониторинг* > *Последние данные*

- *Сбор данных* > *Узлы сети* > *Элементы данных* или *Обнаружение*
- Меню элемента данных
- При редактировании элементов данных или правил низкоуровневого обнаружения (в их формах конфигурации)

Attention:
Проверка должна существовать в кэше конфигурации, чтобы ее можно было выполнить; см. также CacheUpdateFrequency. Кэш **не обновляется** перед выполнением проверки, поэтому последние изменения в конфигурации элемента данных или правила низкоуровневого обнаружения не будут учтены. Чтобы протестировать элементы данных или правила LLD, которые находятся в процессе создания или были созданы недавно, вместо этого можно использовать опцию Тест.

Например, чтобы выполнить пассивную проверку в *Мониторинг* > *Последние данные*:

1. Выберите элементы данных; вы можете выполнить проверку сразу для нескольких элементов.
2. Нажмите *Выполнить сейчас*.



Note:
В *Последних данных* пользователи могут *Выполнить сейчас* элементы данных на узлах сети, для которых у них есть разрешение *чтение-запись*, или разрешение *чтение* и включено действие *Разрешить "Выполнить сейчас" на узлах сети только для чтения* для их роли.

Ограничение проверок агента

Обзор

Вы можете управлять тем, какие ключи элементов данных Zabbix агент или агент 2 разрешено или запрещено использовать при выполнении проверок элементов данных, удаленных команд или скриптов.

Для этого используйте следующие параметры конфигурации агента, чтобы определить правила разрешения/запрета:

- `AllowKey=<pattern>`

- DenyKey=<pattern>
- AllowKeyRegexp=<pattern>
- DenyKeyRegexp=<pattern>

<pattern> должен содержать один ключ элемента данных.
 Для AllowKey и DenyKey <pattern> поддерживает подстановочные знаки (*). Подстановочный знак соответствует любому количеству любых символов в своей позиции и может использоваться для сопоставления ключей элементов данных или параметров (например, vfs.file.*[*]).
 Для AllowKeyRegexp и DenyKeyRegexp <pattern> поддерживает регулярные выражения. Шаблон регулярного выражения можно использовать для сопоставления ключей элементов данных и строк их параметров. См. также примеры корректного экранирования.
 Обратите внимание, что Zabbix agent и Zabbix agent 2 могут поддерживать различный синтаксис регулярных выражений.

Attention:
Для повышения безопасности рекомендуется использовать точные ключи элементов данных вместо подстановочных знаков для AllowKey и DenyKey. Также рекомендуется использовать наиболее специфичные шаблоны регулярных выражений для AllowKeyRegexp и DenyKeyRegexp. Подробности см. в разделе Защита правил разрешения/запрета.

В отличие от других параметров конфигурации агента, вы можете указать неограниченное количество параметров AllowKey, DenyKey, AllowKeyRegexp или DenyKeyRegexp.

Важные примечания

- Все элементы данных system.run по умолчанию отключены (даже если DenyKey и DenyKeyRegexp пусты), как если бы была задана DenyKey=system.run[*] или DenyKeyRegexp=^system\.run\[.*\]$ в качестве последнего правила. Поэтому вы можете разрешить определенные элементы данных system.run, не запрещая явно другие элементы данных system.run.
- По возможности используйте AllowKey/AllowKeyRegexp, чтобы разрешить только необходимые элементы данных и запретить все остальное. Некоторые ключи можно использовать для чтения непредусмотренных файлов через обход пути (например, vfs.file.contents["../../../../etc/passwd"]), а новые версии агента Zabbix могут добавлять ключи, не охваченные вашими правилами DenyKey/DenyKeyRegexp.
- Конфигурация AllowKey, DenyKey, AllowKeyRegexp и DenyKeyRegexp не влияет на параметры агента HostnameItem, HostMetadataItem или HostInterfaceItem.
- Запрещенные элементы данных становятся неподдерживаемыми без каких-либо подсказок или сообщений об ошибках; например:
  - Команда агента Zabbix --print (-p) в командной строке не будет показывать запрещенные ключи элементов данных.
  - Команда агента Zabbix --test (-t) в командной строке будет возвращать "Unsupported item key." для запрещенных ключей элементов данных.
  - При включенном журналировании (LogRemoteCommands=1) файл журнала агента Zabbix не будет записывать запрещенные удаленные команды.

Порядок правил allow/deny

Вы можете указать неограниченное количество правил AllowKey, DenyKey, AllowKeyRegexp или DenyKeyRegexp, однако их порядок имеет значение.

- Правила проверяются по одному, сверху вниз.
- Когда ключ элемента данных соответствует правилу, он либо разрешается, либо запрещается, и проверка правил прекращается.

Например, при проверке vfs.file.contents[/etc/passwd] правила обрабатываются следующим образом:

```
AllowKey=vfs.file.contents[/tmp/app.log]    #                    ,                    .
AllowKey=vfs.file.contents[/etc/passwd]     #                  ;
DenyKey=vfs.file.*[*]                       #            ,                .
```

Следующий порядок правил запретит проверку элемента данных:

```
DenyKey=vfs.file.*[*]                       #                  ;
AllowKey=vfs.file.contents[/etc/passwd]     #            ,                .
AllowKey=vfs.file.contents[/tmp/app.log]    #            ,                .
```

Примеры

Следующие примеры показывают распространенные шаблоны конфигурации для AllowKey, DenyKey, AllowKeyRegexp и DenyKeyRegexp.

Разрешение определенных проверок и команд

Разрешите только две проверки элементов данных vfs.file и две команды system.run:

с использованием AllowKey и DenyKey:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

с использованием AllowKeyRegexp и DenyKeyRegexp:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Указывать DenyKey=system.run[*] или DenyKeyRegexp=^system\.run\[.*\]$ не требуется, поскольку все остальные команды system.run по умолчанию запрещены.

Разрешение выполнения скриптов

Разрешите агенту Zabbix выполнять скрипты на узлах сети всеми доступными способами:

- Глобальные скрипты, которые могут выполняться во веб-интерфейсе или через API (в этом случае всегда используется ключ system.run[myscript.sh])
- Удаленные команды из операций действия (в этом случае всегда используется ключ system.run[myscript.sh,nowait])
- Элементы данных system.run агента Zabbix со скриптом, например:
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

или

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Чтобы управлять параметром wait/nowait, необходимо задать другое правило. Например, можно разрешить только элементы данных system.run[myscript.sh,wait], тем самым исключив другие способы:

```
AllowKey=system.run[myscript.sh,wait]
```

или

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Защита правил allow/deny

Эти примеры показывают, как защитить слишком разрешающие правила AllowKey/AllowKeyRegexp или DenyKey/DenyKeyRegexp.

Рассмотрим следующие правила:

с использованием AllowKey и DenyKey:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

с использованием AllowKeyRegexp и DenyKeyRegexp:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
В Windows необходимо экранировать пробелы в пути с помощью символа каретки (^) для AllowKey или DenyKey и \s для AllowKeyRegexp или DenyKeyRegexp.

Эти правила содержат подстановочный знак (*) (или .* для шаблонов на основе регулярных выражений), который может быть использован неправомерно:

- Скрипт test.bat можно выполнить с любыми аргументами, включая нежелательные.
- Шаблон vfs.file.* (^vfs\.file\..*$) соответствует ключам элементов данных как с параметрами, так и без них; однако все элементы данных vfs.file требуют параметров.
- Шаблон system.cpu.load[*] (^system\.cpu\.load\[.*\]$) соответствует только ключам элементов данных с параметрами; однако элементы данных system.cpu.load не требуют параметров.

Чтобы защитить эти правила, явно разрешите выполнение test.bat только с определенными аргументами и запретите корректные шаблоны ключей элементов данных; например:

с использованием AllowKey и DenyKey:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

с использованием AllowKeyRegexp и DenyKeyRegexp:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

Вы можете проверить правила, выполнив следующие команды, которые вернут ZBX_NOTSUPPORTED.

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

Примеры шаблонов

Следующие таблицы показывают, как сопоставляются шаблоны ключей элементов данных:

- Ключ соответствует шаблону только в том случае, если выполняются **все** условия в столбце *Matches*.
- Параметры должны быть полностью заключены в квадратные скобки (например, vfs.file.contents[* и vfs.file.contents*utf8] являются недопустимыми шаблонами).
- Примеры иллюстрируют только сопоставление шаблонов; фактические элементы данных требуют параметров.

Для AllowKey и DenyKey:

Pattern	Matches	Examples
*	Любой ключ с параметрами или без них	
vfs.file.*	Ключ начинается с vfs.file. С параметрами или без них	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	Ключ начинается с vfs. Ключ заканчивается на .contents Без параметров	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]

Pattern	Matches	Examples
vfs.file.*[*]	Ключ начинается с vfs.file. Любые параметры или пустые параметры	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
vfs.file.contents	Ключ — vfs.file.contents Без параметров	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	Ключ — vfs.file.contents[] Пустые параметры	Matches: vfs.file.contents[] Does not match: vfs.file.contents
vfs.file.contents[*]	Ключ — vfs.file.contents Любые параметры или пустые параметры	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents
vfs.file.contents[/etc/passwd,*]	Ключ — vfs.file.contents Первый параметр — /etc/passwd Любой второй параметр или пустой	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	Ключ — vfs.file.contents Любые параметры, как минимум один содержит passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	Ключ — vfs.file.contents Первый параметр содержит passwd Любой второй параметр или пустой	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	Ключ — vfs.file.contents Первый параметр — /etc/passwd Второй параметр — utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Для AllowKeyRegexp и DenyKeyRegexp:

Pattern	Matches	Examples
^.*$	Любой ключ с параметрами или без них	
^vfs\.file\..*$	Ключ начинается с vfs.file. С параметрами или без них	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]

Pattern	Matches	Examples
^vfs\..*\.contents$	Ключ начинается с vfs. Ключ заканчивается на .contents Без параметров	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	Ключ начинается с vfs.file. Любые параметры или пустые параметры	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	Ключ — vfs.file.contents Без параметров	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[\]$	Ключ — vfs.file.contents[] Пустые параметры	Matches: vfs.file.contents[] Does not match: vfs.file.contents
^vfs\.file\.contents\[.*\]$	Ключ — vfs.file.contents Любые параметры или пустые параметры	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents
^vfs\.file\.contents\[/etc/passwd,.*\]$	Ключ — vfs.file.contents Первый параметр — /etc/passwd Любой второй параметр или пустой	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[.*passwd.*\]$	Ключ — vfs.file.contents Любые параметры, как минимум один содержит passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
^vfs\.file\.contents\[.*passwd.*,.*\]$	Ключ — vfs.file.contents Первый параметр содержит passwd Любой второй параметр или пустой	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$	Ключ — vfs.file.contents Первый параметр — /etc/passwd Второй параметр — utf8 или windows-1252	Matches: vfs.file.contents[/etc/passwd,utf8] vfs.file.contents[/etc/passwd,windows-1252] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Обнаружение проблем с помощью триггеров

Обнаружение проблем в Zabbix основано на **триггерах**.

Триггер описывает условие проблемы, позволяя Zabbix реагировать, если проблема возникает.

Полный рабочий процесс обнаружения проблем выглядит так:

1. Убедитесь, что у вас есть элемент данных, собирающий данные.
2. Определите триггер для этого элемента данных. Без триггера Zabbix только собирает данные, но никогда не реагирует на них.
3. Просматривайте обнаруженные проблемы:
   - В *Monitoring* > *Problems*
   - В виджете *Problems*
4. Получайте уведомления, выполняйте удаленные команды (если они определены).

Обратите внимание, что официальные шаблоны Zabbix содержат заранее определенные триггеры.
См. руководство по применению шаблона для monitor Linux.

Что такое ”триггер”?

Триггер описывает *условие проблемы* с помощью выражения триггера.

Условие проблемы, когда оно возникает в реальной жизни, означает, что триггер ”срабатывает” (в Zabbix создаётся проблема).

Условие проблемы может основываться на последнем значении, среднем значении, обнаруженной строке и многих других функциях. Триггеры нельзя создавать для элементов данных, возвращающих значения с типом данных binary или JSON.

Простое выражение триггера вычисляет функцию (например, `max()` для максимального значения) для данных, указанных в скобках (обычно это элемент данных и временное окно), а затем сравнивает результат вычисления с пороговым значением.

Например:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Этот триггер срабатывает, если свободное место на диске, измеряемое в течение 5 минут, постоянно было ниже 10 ГБ.

Триггер может иметь следующие состояния:

Состояние триггера	Описание	В Zabbix
OK	Выражение триггера вычислено как ”0” (или FALSE)	Проблема не создана Проблема устранена (если существовала)
Проблема	Выражение триггера вычислено как ”1” (или TRUE)	Проблема создана

Иногда вычисление выражения триггера по какой-либо причине невозможно. См. Неизвестное состояние выражения.

Другие ресурсы:

- Примеры триггеров
- Настройка триггера
- Рекомендации

Частота вычисления

Триггер пересчитывается каждый раз, когда сервер Zabbix получает **новое значение**, которое входит в выражение.
При получении нового значения пересчитывается каждая функция, включенная в выражение (а не только та, которая получила новое значение).

Кроме того, триггер пересчитывается каждые 30 секунд (history syncer), если выражение содержит:

- функции даты и времени
- функцию nodata()

Триггеры, содержащие только функции trend, вычисляются один раз за наименьший временной период в выражении. Хотя многие функции триггеров используют данные history значений элемента данных, функции trend используют данные трендов.

Примеры триггеров

На этой странице собрана коллекция примеров триггеров.

Примеры отсортированы по варианту использования:

- Недоступный узел сети
- Недоступный прокси
- Недоступный кластер SMTP
- Недоступный узел сети
- Неожиданная перезагрузка
- Изменения в важных файлах
- Изменения в DNS-запросе
- Несовпадающее ПО на разных узлах сети
- Несинхронизированные часы
- Устаревший агент
- Флаппинг интерфейса
- Высокий входящий трафик
- Мало свободного места на диске
- Мало свободного места на диске (динамический порог)
- Высокая загрузка CPU
- Сравнение загрузки CPU
- Сравнение загрузки CPU за длительный период
- Строка, содержащая специальные символы

Обнаружение недоступного узла сети

```
max(/host/zabbix[host,agent,available],5m)=0
```

Этот триггер срабатывает, если Zabbix агент на узле сети недоступен в течение 5 минут.
 Функция: max
 Элемент данных: zabbix[host,agent,available]

Альтернатива:

```
nodata(/host/agent.ping,5m)=1
```

Этот триггер срабатывает, если от Zabbix агента не поступали данные в течение 5 минут.
 Функция: nodata
 Элемент данных: agent.ping

Обнаружение недоступного прокси

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Этот триггер срабатывает, если данные прокси Zabbix отстают от времени сервера Zabbix на 1 минуту.
 Функция: fuzzytime
 Элемент данных: zabbix[proxy,{$PROXY_NAME},lastaccess]

Обнаружение недоступного кластера SMTP

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Этот триггер срабатывает, если оба SMTP-сервера недоступны.
 Функция: last
 Элемент данных: net.tcp.service

Обнаружение недоступного узла сети

```
count(/host/icmpping,30m,,"0")>5
```

Этот триггер срабатывает, если узел сети недоступен по ping более 5 раз за последние 30 минут.
 Функция: count
 Элемент данных: icmpping

Обнаружение неожиданной перезагрузки

```
change(/host/system.uptime)<0
```

Этот триггер срабатывает, если обнаружено отрицательное изменение значения времени работы системы (что указывает на перезагрузку).
 Функция: change
 Элемент данных: system.uptime

Обнаружение изменений в важных файлах

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Этот триггер срабатывает, если файл /etc/passwd был изменен. Выражение истинно, когда предыдущая контрольная сумма /etc/passwd отличается от самой последней. Подобные выражения могут быть полезны для мониторинга

изменений в важных файлах, таких как /etc/passwd, /etc/inetd.conf, /kernel и т. д.
 Функция: last
 Элемент данных: vfs.file.cksum

Обнаружение изменений в DNS-запросе

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Обратите внимание на кавычки вокруг второго операнда.

Этот триггер срабатывает, если результат запроса не равен тому, что он обычно возвращает:

```
example.com           MX       0 mail.example.com
```

Функция: last
 Элемент данных: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], с макросами, определенными следующим образом:

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Обнаружение несовпадающего программного обеспечения на разных узлах сети

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Этот триггер срабатывает, если версия Ubuntu отличается на разных узлах сети. Обратите внимание, что операндами здесь являются функции, которые возвращают строки.
 Функция: last
 Элемент данных: vfs.file.contents

Обнаружение несинхронизированных часов

```
fuzzytime(/host/system.localtime,10s)=0
```

Триггер срабатывает, если локальное время клиента и время сервера Zabbix отличаются более чем на 10 секунд.
 Функция: fuzzytime
 Элемент данных: system.localtime

Обратите внимание, что system.localtime должен быть настроен как пассивная проверка для агента Zabbix; в Zabbix agent 2 он может быть настроен как активная проверка.

Обнаружение устаревшего агента

```
find(/host/agent.version,,"like","beta")=1
```

Этот триггер срабатывает, если Zabbix агент имеет бета-версию. Zabbix агент необходимо обновить.
 Функция: find
 Элемент данных: agent.version

Обнаружение нестабильного интерфейса

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Этот триггер срабатывает, если рабочее состояние (up/down/unknown) eth0 изменилось более 5 раз в течение часа.
 Функция: changecount
 Элемент данных: vfs.file.contents

Обнаружение высокого входящего трафика

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Этот триггер срабатывает, если количество полученных байтов на eth0 за последние пять минут всегда было больше 100 килобайт. Вероятно, кто-то загружает большой файл.
 Функция: min
 Элемент данных: net.if.in[eth0,bytes]

Обнаружение нехватки места на диске

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Триггер срабатывает, если свободное место на диске стабильно (в течение 5 минут) ниже 10 ГБ.

Вы также можете определить выражение восстановления:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

Проблема **устраняется** только тогда, когда свободное место на диске стабильно (в течение 10 минут) превышает 40 ГБ.
 Функция: min
 Элемент данных: vfs.fs.size

Обнаружение нехватки места на диске (динамический порог)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

Триггер срабатывает, если свободное пространство хранилища (в единицах выделения) падает ниже 10 процентов. Обратите внимание на использование значения другого элемента данных для получения адаптивного порога триггера, применимого к обнаруженным хранилищам различного размера. Функция: last

Обнаружение высокой загрузки CPU

```
last(/host/system.cpu.load[all,avg1])>5
```

Триггер срабатывает, когда средняя загрузка процессора превышает 5 в течение одной минуты.

Варианты:

```
min(/host/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

```
min(/host/system.cpu.load[all,avg1],5m)>2 and not (dayofweek()=7 and time()>230000) and not (dayofweek()=1
```

Такие триггеры анализируют данные за 5 минут и срабатывают, только если загрузка CPU ни разу не опускается ниже 2. Дополнительно эти триггеры срабатывают:

- только в ночное время (00:00-06:00)
- в любое время, кроме 2 часов при смене недели (воскресенье, 23:00 - понедельник, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Этот триггер срабатывает, если загрузка процессора слишком высока как минимум на двух из трех узлов сети.
 Функции: last, min, dayofweek, time
 Элемент данных: system.cpu.load

Сравнение загрузки CPU

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

Триггер срабатывает, если средняя загрузка сегодня более чем в два раза превышает среднюю загрузку за тот же час вчера (с использованием сдвига времени now-1d).
 Функция: avg
 Элемент данных: system.cpu.load

Сравнение долгосрочной загрузки CPU

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Этот триггер срабатывает, если загрузка CPU на узле сети за последний месяц увеличилась более чем на 10%
 Функция: trendavg
 Элемент данных: system.cpu.load

Вы также можете использовать поле Имя события в настройках триггера, чтобы сформировать осмысленное сообщение оповещения, например, чтобы получить что-то вроде

```
"           Exchange           24%      (0.69)               (0.56)"
```

имя события должно быть определено как:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

Для такого рода проблемы также полезно разрешить ручное закрытие в настройках триггера.

Обнаружение строки, содержащей специальные символы

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Триггер срабатывает, если содержимое /tmp/hello равно строке, определенной в {$HELLO_MACRO}:

```
{$HELLO_MACRO} = \" //hello ?\"
```

В качестве альтернативы можно сравнить со строкой напрямую:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Обратите внимание, как специальные символы (\\ и ") экранируются, когда строка сравнивается напрямую.

Функция: last
 Элемент данных: vfs.file.contents

Настройка триггера

На этой странице описывается, как настроить триггер во веб-интерфейсе Zabbix.

Обратите внимание, что официальные шаблоны Zabbix содержат предопределенные триггеры.
См. руководство по применению шаблона для мониторинга Linux.
Также можно выполнить массовое обновление настроенных триггеров.

Другие ресурсы:

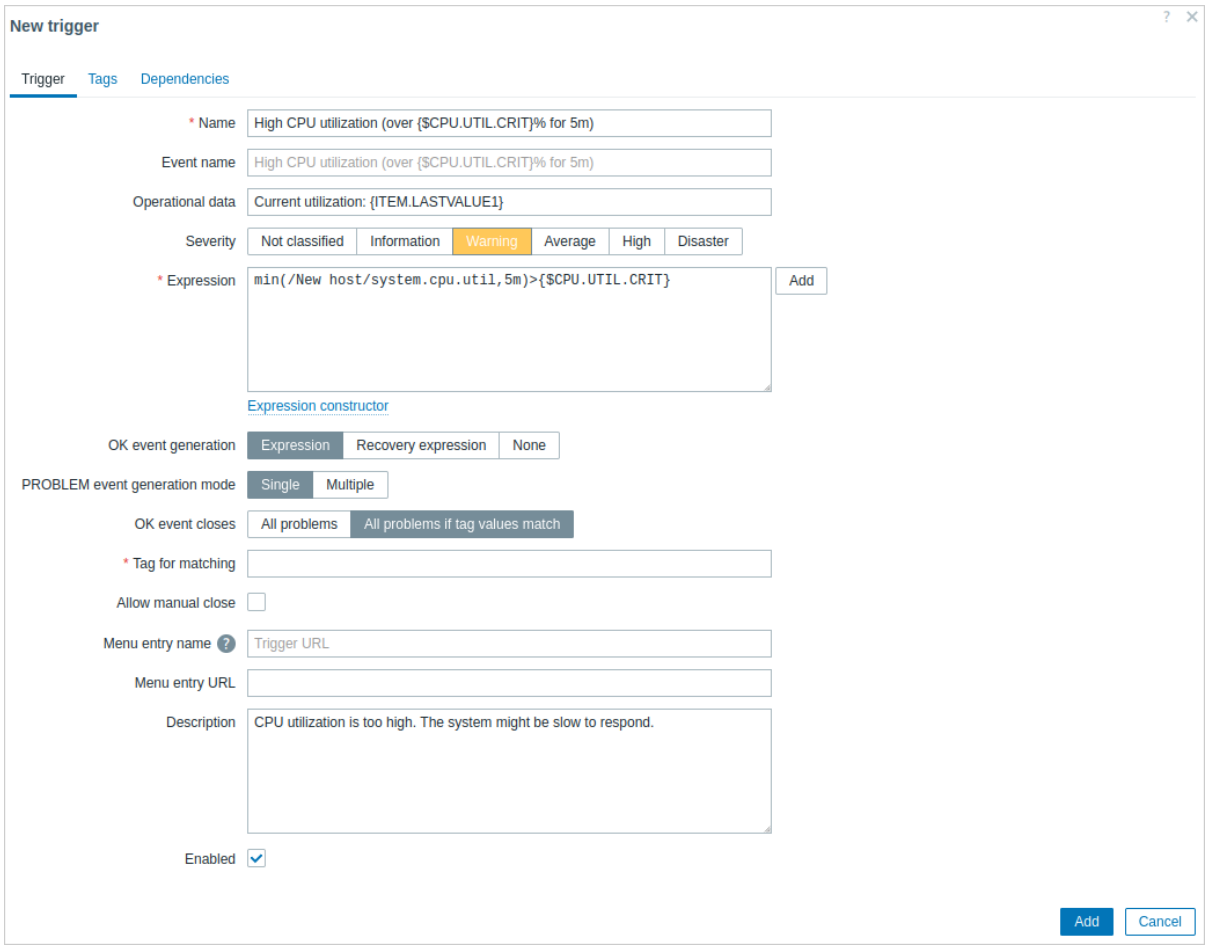
- Обнаружение проблем с помощью триггеров (общее введение)
- Примеры триггеров (по вариантам использования)
- Выражение триггера и функции (подробности синтаксиса)
- Рекомендации по лучшим практикам

Конфигурация

Чтобы настроить триггер, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Нажмите *Триггеры* в строке узла сети.
3. Нажмите *Создать триггер* справа (или имя триггера, чтобы изменить существующий триггер).
4. Введите параметры триггера в форме.

Вкладка **Trigger** содержит все основные атрибуты триггера.



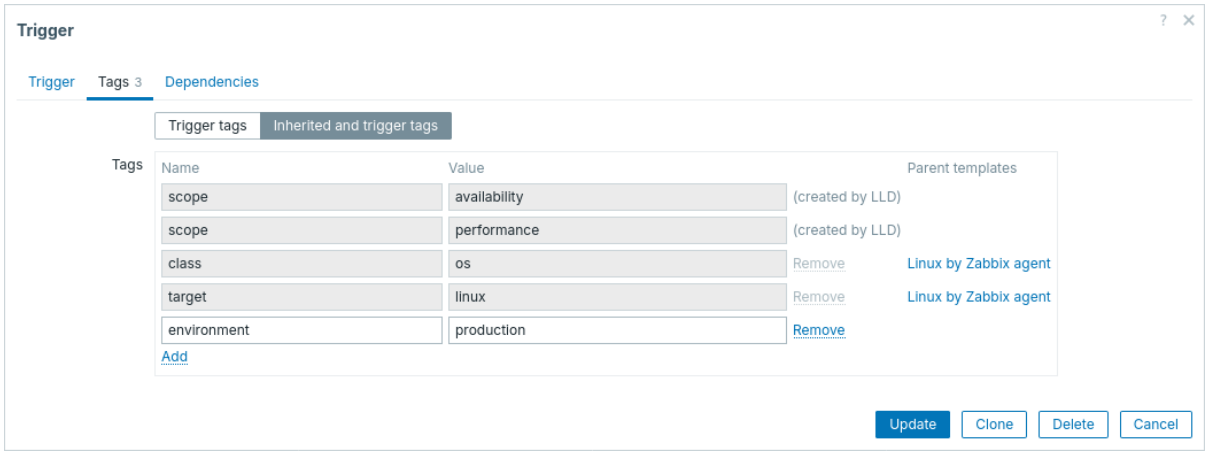
Все обязательные поля ввода отмечены красной звездочкой.

Parameter	Description
Name	Имя триггера. Поддерживаются макросы: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*} и пользовательские макросы {$MACRO}. Макросы **$1, $2...$9** можно использовать для ссылки на первую, вторую... девятую константу выражения. *Примечание*: макросы $1-$9 будут корректно разрешаться, если они ссылаются на константы в относительно простых, прямолинейных выражениях. Например, имя "Processor load above $1 on {HOST.NAME}" автоматически изменится на "Processor load above 5 on New host", если выражение last(/New host/system.cpu.load[percpu,avg1])>5.

Parameter	Description
Event name	Если задано, это имя будет использоваться для создания имени события проблемы. По умолчанию имя события совпадает с именем триггера. Имя события можно использовать для формирования информативных оповещений, содержащих данные о проблеме (см. пример). Поддерживается тот же набор макросов, что и для имени триггера, плюс макросы выражения {TIME}, {TIMESTAMP} и {?EXPRESSION}.
Operational data	Введите строку с макросами для отображения динамических данных в реальном времени в *Мониторинг* > *Проблемы*. Либо оставьте это поле пустым, чтобы отображались последние значения всех элементов данных из выражения триггера. Поддерживается тот же набор макросов, что и для имени триггера, с возможностью динамического разрешения. Например: {ITEM.VALUE<1-9>} разрешается в значения элементов данных в момент изменения состояния триггера (создание проблемы, восстановление, ручное закрытие или закрытие по корреляции). {ITEM.LASTVALUE<1-9>} разрешается в последние значения элементов данных. Обратите внимание, что при ручном закрытии проблемы новое значение не создается, поэтому оба макроса по-прежнему будут показывать значение на момент возникновения проблемы. Также обратите внимание, что оба макроса разрешаются в *UNKNOWN*, если последнее значение старше *Max history display period* (см. *Администрирование* > *Общие*).
Severity	Установите требуемую серьезность триггера, нажимая кнопки.
Expression	Логическое выражение, используемое для определения условий возникновения проблемы. Поддерживаются суффиксы времени и суффиксы размера памяти. Проблема создается после выполнения всех условий, включенных в выражение, то есть когда выражение принимает значение TRUE. Проблема будет восстановлена, как только выражение станет FALSE, если только дополнительные условия восстановления не указаны в *Recovery expression*.
OK event generation	Параметры генерации событий OK: **Expression** - события OK генерируются на основе того же выражения, что и события проблемы; **Recovery expression** - события OK генерируются, если выражение проблемы принимает значение FALSE, а выражение восстановления - TRUE; **None** - в этом случае триггер никогда не вернется в состояние OK самостоятельно.
Recovery expression	Логическое выражение (необязательно), определяющее дополнительные условия, которые должны быть выполнены до восстановления проблемы, после того как исходное выражение проблемы уже было оценено как FALSE. Выражение восстановления полезно для гистерезиса триггера. Невозможно восстановить проблему только по выражению восстановления, если выражение проблемы по-прежнему TRUE. Это поле доступно только если для *OK event generation* выбрано 'Recovery expression'.
PROBLEM event generation mode	Режим генерации событий проблемы: **Single** - одно событие генерируется, когда триггер впервые переходит в состояние 'Problem'; **Multiple** - событие генерируется при *каждой* оценке триггера как 'Problem'.
OK event closes	Выберите, закрывает ли событие OK: **All problems** - все проблемы этого триггера; **All problems if tag values match** - только те проблемы триггера, у которых совпадают значения тегов событий.
Tag for matching	Введите имя тега события, которое будет использоваться для корреляции событий. Это поле отображается, если для свойства *OK event closes* выбрано 'All problems if tag values match', и в этом случае является обязательным.
Allow manual close	Установите этот флажок, чтобы разрешить ручное закрытие событий проблемы, создаваемых этим триггером. Ручное закрытие возможно при подтверждении событий проблемы.
Menu entry name	Если поле не пустое, введенное здесь имя (до 64 символов) используется в нескольких местах веб-интерфейса как подпись для URL триггера, указанного в параметре *Menu entry URL*. Если поле пустое, используется имя по умолчанию *Trigger URL*. Поддерживается тот же набор макросов, что и для имени триггера, плюс {EVENT.ID}, {HOST.ID} и {TRIGGER.ID}.

Parameter	Description
Menu entry URL	Если поле не пустое, введенный здесь URL (до 2048 символов) доступен как ссылка в меню события в нескольких местах веб-интерфейса, например при нажатии на имя проблемы в виджете панели *Мониторинг* > *Проблемы* или *Проблемы*. Поддерживается тот же набор макросов, что и для имени триггера, плюс {EVENT.ID}, {HOST.ID} и {TRIGGER.ID}. Примечание: пользовательские макросы со скрытыми значениями не будут разрешены в URL.
Description	Текстовое поле для дополнительной информации об этом триггере. Может содержать инструкции по устранению конкретной проблемы, контактные данные ответственных сотрудников и т. д. Поддерживается тот же набор макросов, что и для имени триггера.
Enabled	Снимите этот флажок, чтобы при необходимости отключить триггер. Проблемы отключенного триггера больше не отображаются в веб-интерфейсе, но не удаляются.

Вкладка **Tags** позволяет определить теги на уровне триггера. Все проблемы этого триггера будут помечены значениями, введенными здесь.



Кроме того, параметр *Inherited and trigger tags* позволяет просматривать теги, определенные на уровне шаблона, если триггер получен из этого шаблона. Если существует несколько шаблонов с одинаковым тегом, эти теги отображаются один раз, а имена шаблонов разделяются запятыми. Триггер не ”наследует” и не отображает теги уровня узла сети.

Parameter	Description
Name/Value	Задайте пользовательские теги для пометки событий триггера. Теги представляют собой пару имени тега и значения. Можно использовать только имя или указать его вместе со значением. Триггер может иметь несколько тегов с одинаковым именем, но разными значениями. В тегах событий поддерживаются пользовательские макросы, пользовательские макросы с контекстом, макросы низкоуровневого обнаружения и макросы functions с `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}`. Макросы низкоуровневого обнаружения можно использовать внутри контекста макроса. Макрос {TRIGGER.ID} поддерживается в значениях тегов триггера. Это может быть полезно для идентификации триггеров, созданных из прототипов триггеров, и, например, для подавления проблем от этих триггеров во время обслуживания. Если общая длина развернутого значения превышает 255, оно будет обрезано до 255 символов. См. все макросы, поддерживаемые для тегов событий. Теги событий можно использовать для корреляции событий, в условиях действий, а также они будут видны в *Мониторинг* > *Проблемы* или в виджете *Проблемы*.

Вкладка **Dependencies** содержит все зависимости триггера.

Нажмите *Add*, чтобы добавить новую зависимость.

Note:
Вы также можете настроить триггер, открыв существующий, нажав кнопку *Clone*, а затем сохранив его под другим именем.

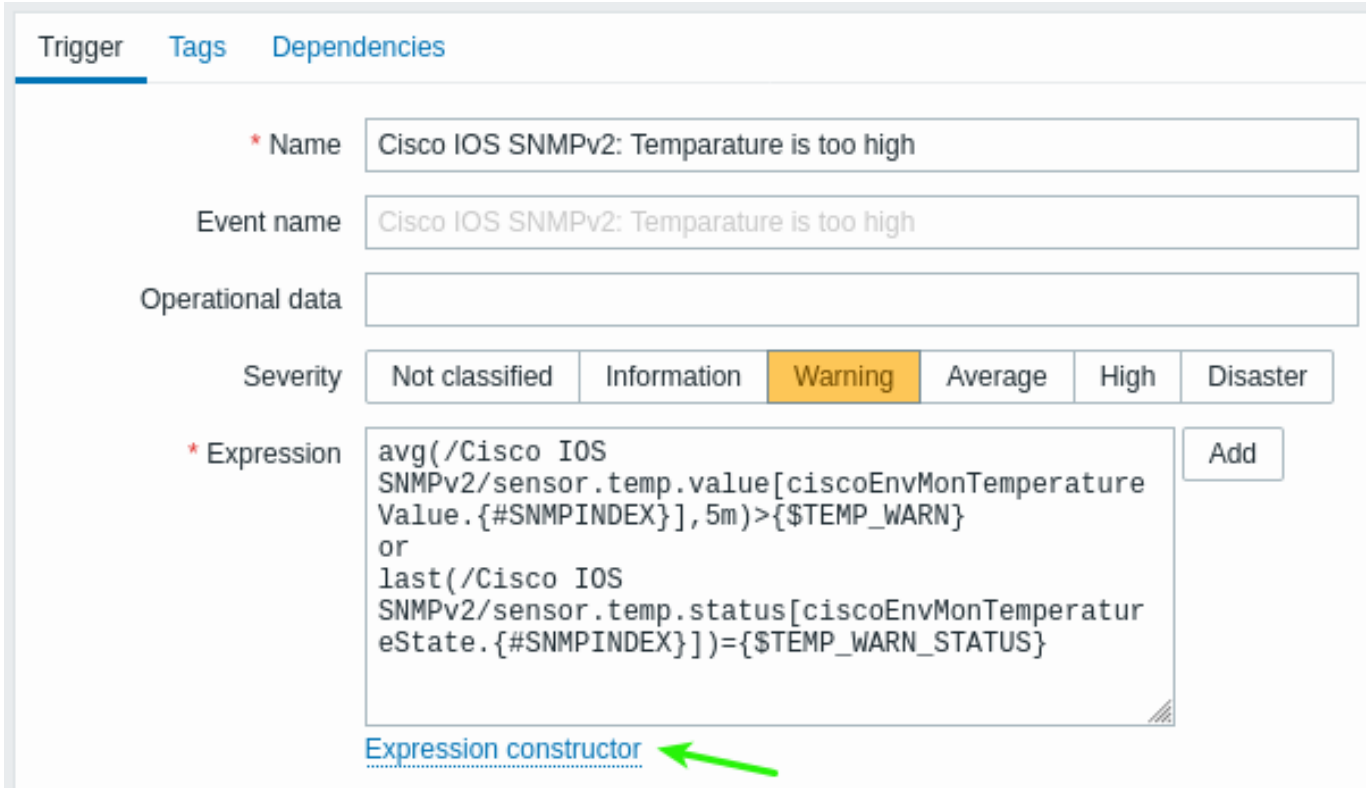
Тестирование выражения триггера

Можно проверить, каким будет результат выражения в зависимости от полученного значения.

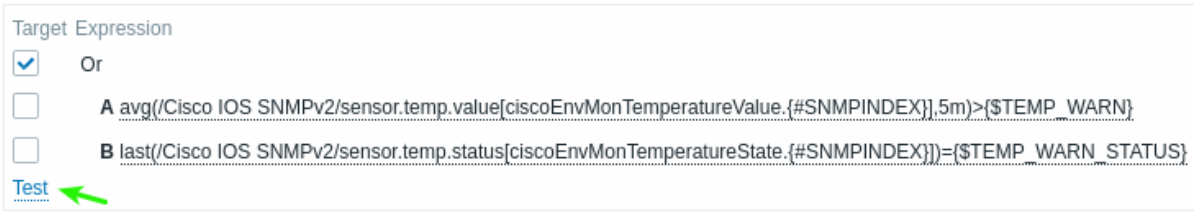
В качестве примера взято следующее выражение из официального шаблона:

```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],5m)>{$TEMP_WARN}
or
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}])={$TEMP_WARN_STATUS}
```

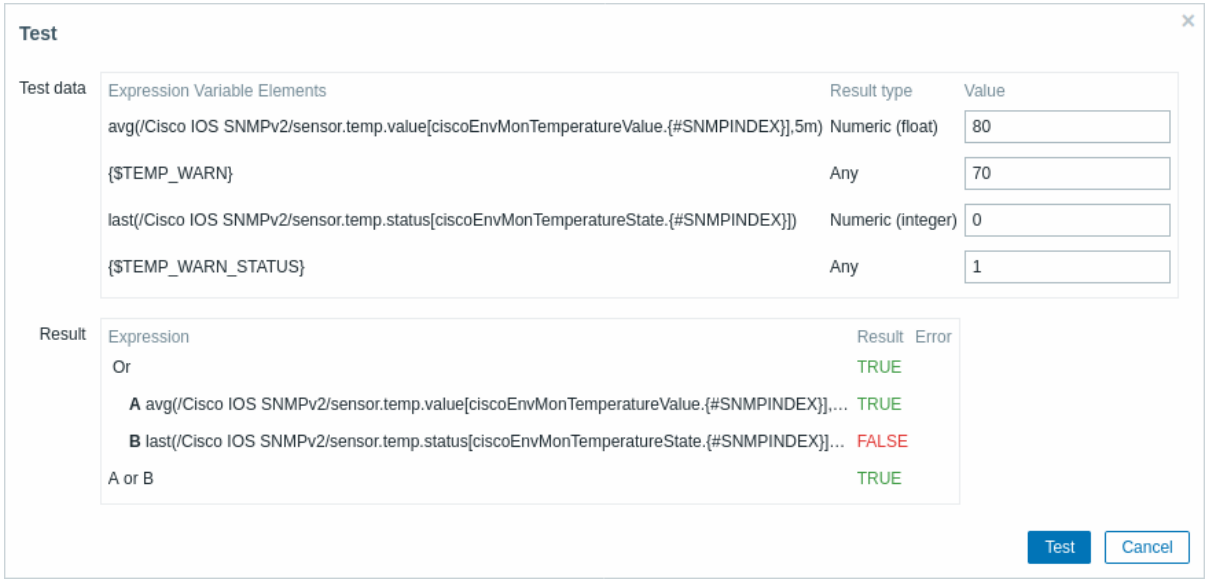
Чтобы протестировать выражение, нажмите *Конструктор выражения* под полем выражения.



В Конструкторе выражения перечислены все отдельные выражения. Чтобы открыть окно тестирования, нажмите *Тест* под списком выражений.



В окне тестирования можно ввести примерные значения (’80’, ’70’, ’0’, ’1’ в этом примере), а затем посмотреть результат выражения, нажав кнопку *Тест*.



Можно увидеть результат как отдельных выражений, так и всего выражения целиком.

”TRUE” означает, что указанное выражение верно. В данном случае A значение ”80” больше указанного значения `{$TEMP_WARN}`, которое в этом примере равно ”70”. Как и ожидалось, отображается результат ”TRUE”.

”FALSE” означает, что указанное выражение неверно. В данном случае B значение `{$TEMP_WARN_STATUS}` ”1” должно быть равно указанному значению, ”0” в этом примере. Как и ожидалось, отображается результат ”FALSE”.

Выбранный тип выражения — ”OR”. Если хотя бы одно из указанных условий (в данном случае A или B) имеет значение TRUE, итоговый результат также будет TRUE. Это означает, что текущее значение превышает порог предупреждения и возникла проблема.

Выражение триггера и функции

Обзор

На этой странице описываются синтаксис выражения триггера и детали его вычисления.

Синтаксис простого выражения:

`function(/host/item,time_period)<operator><constant>`

В этом выражении первый операнд (слева от оператора) — это функция с ее параметрами в круглых скобках (обычно это элемент данных и период времени).

Функция используется для анализа полученных данных за заданный период времени, в результате чего вычисляется значение.

Затем это значение сравнивается со вторым операндом с использованием оператора. В этом примере второй операнд — константа, но им также может быть другая функция.

Например:

`min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K`

Этот триггер срабатывает, если количество полученных байтов на `eth0` за последние пять минут всегда было выше 100 килобайт. В этом случае выражение истинно, и создается проблема.

Выражения триггеров чрезвычайно гибкие. В более сложных выражениях можно комбинировать несколько функций, операторов и констант.

См. также:

- Примеры триггеров
- Обнаружение проблем с помощью триггеров (общее введение)

Функции

Функции позволяют анализировать собранные значения, например вычислять среднее значение или находить определенную строку.

Нажмите соответствующую группу функций, чтобы просмотреть подробности.

Группа функций	Функции
Агрегатные функции	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Функции Foreach	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Побитовые функции	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Функции даты и времени	date, dayofmonth, dayofweek, now, time
Функции истории	change, changecount, count, countunique, find, first, firstclock, fuzzytime, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
Функции трендов	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
Математические функции	abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
Функции операторов	between, in
Прогнозирующие функции	forecast, timeleft
Строковые функции	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Если не указано иное, эти функции поддерживаются в следующих компонентах:

- Выражения триггеров
- Вычисляемые элементы данных
- Макросы выражений

Функции Foreach поддерживаются только для агрегатных вычислений.

Обычно функции возвращают числовые значения для сравнения. При возврате строк сравнение возможно с операторами **=** и **<>** (см. пример Обнаружение несовпадающего программного обеспечения на разных узлах сети).

Параметры функций

Параметры функций позволяют указать:

- ключ элемента данных (в формате `/host/key`) для функций, ссылающихся на историю элемента данных узла сети
- период времени (и другие параметры, зависящие от функции)
- другие выражения

Ключ элемента данных

Ссылающийся элемент данных должен находиться в поддерживаемом состоянии (кроме функции **nodata()**, которая вычисляется также и для неподдерживаемых элементов данных).

Опускать имя узла сети в первом параметре (то есть как в `function(/key,parameter,...)`) поддерживается только в определённых контекстах:

- В формуле вычисляемых элементов данных
- В макросах выражений, которые могут использоваться в:
  - поле *Имя события*
  - имени графика
  - метке элементов карты "Host" и "Trigger" map elements

В этих контекстах вы также можете использовать макрос `{HOST.HOST}`. `{HOST.HOST<1-9>}` можно использовать в случае поля *Имя события* и элемента карты "Trigger" для ссылки на конкретный элемент данных в выражении триггера. Если в этих контекстах имя узла сети опущено или заменено на `{HOST.HOST}`, ссылка указывает на первый элемент данных в выражении триггера или на первый элемент данных на графике. Вне этих поддерживаемых контекстов опускание имени узла сети в выражениях триггера приведёт к ошибке. См. пример Compare long-term CPU loads для иллюстрации использования двойной косой черты в макросах имени события.

Период времени

Параметры, зависящие от функции, указываются после ключа элемента данных и отделяются от него запятой.

Большинство числовых функций принимает **период времени** в качестве параметра. Он позволяет указать интересующий нас интервал. Его можно задать как период времени (30s, 10m, 1h) или как диапазон значений (#5 - пять последних значений).

Для указания периода времени можно использовать секунды или суффиксы времени. Если перед параметром указать знак решетки, он будет иметь другое значение:

Expression	Description
sum(/host/key,**10m)**	Сумма значений за последние 10 минут.
sum(/host/key,**#10)**	Сумма последних десяти значений.

Параметры со знаком решетки имеют другое значение для функции **last** - они обозначают N-е предыдущее значение. Например, для значений 30, 70, 20, 60, 50 (от самого последнего к самому раннему):

- `last(/host/key,#2)` вернет «70»
- `last(/host/key,#5)` вернет «50»

Период времени отсчитывается до «текущего момента» - то есть до последнего времени перерасчета триггера (см. раздел Частота вычисления); «текущий момент» не соответствует текущему времени сервера.

Период времени определяет одно из следующего:

- Все значения между «текущим моментом - период времени» и «текущим моментом» (или, если применяется смещение времени, между «текущим моментом - смещение времени - период времени» и «текущим моментом - смещение времени»).
- Указанное количество прошлых значений вплоть до «текущего момента».

Если за указанный период оценки отсутствуют данные, триггер или вычисляемый элемент данных, использующий эту функцию, становится неподдерживаемым.

Обратите внимание:

- Если в триггере используется только одна функция (ссылающаяся на историю данных), «текущий момент» всегда соответствует времени получения последнего значения. Например, если последнее значение было получено час назад, период времени будет отсчитываться до последнего значения, полученного час назад.
- Новый триггер вычисляется сразу после получения первого значения (для функций истории); для функций даты и времени и nodata() вычисление выполняется в течение 30 секунд. Таким образом, триггер будет вычислен, даже если с момента его создания еще не прошел указанный период времени (например, один час). Триггер также будет вычислен после получения первого значения, даже если диапазон времени задан, например, как десять последних значений.

Смещение времени

Поддерживается необязательное смещение времени с указанием времени или количества значений в качестве параметра функции. Этот параметр позволяет ссылаться на данные за прошедший период времени.

Смещение времени начинается с `now`, обозначающего текущее время, за которым следует `+N<` > или `-N<` > для добавления или вычитания N единиц времени.

Например, `avg(/host/key,1h:now-1d)` вернет среднее значение за час, прошедший день назад.

> **Attention:**
> Смещение времени, указанное в месяцах (M) и годах (y), поддерживается только для функций трендов. Другие функции поддерживают секунды (s), минуты (m), часы (h), дни (d) и недели (w).

Смещение времени с абсолютными периодами времени

В параметре смещения времени поддерживаются абсолютные периоды времени, например с полуночи до полуночи для дня, с понедельника по воскресенье для недели, с первого по последний день месяца для месяца.

Смещение времени для абсолютных периодов времени начинается с `now`, обозначающего текущее время, за которым может следовать любое количество операций со временем: `/<` > определяет начало и конец единицы времени, например период с полуночи до полуночи для дня; `+N<` > или `-N<` > добавляет или вычитает N единиц времени.

Обратите внимание, что значение смещения времени может быть больше или равно 0, тогда как минимальное значение периода времени равно 1.

Параметр	Описание
1d:now/d	Вчера
1d:now/d+1d	Сегодня
2d:now/d+1d	Последние 2 дня
1w:now/w	Прошлая неделя
1w:now/w+1w	Эта неделя

Другие выражения

Параметры функций могут содержать другие выражения, как в следующем синтаксисе:

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Обратите внимание, что другие выражения не могут использоваться, если функция ссылается на историю элемента данных. Например, следующий синтаксис **не** допускается:

~~`min(/host/key,#5*10)`~~

Хотя другие выражения триггера в качестве параметров функций ограничены функциями без истории в триггерах, это ограничение не применяется к вычисляемым элементам данных.

Операторы

Для триггеров поддерживаются следующие операторы **(в порядке убывания приоритета выполнения)**:

Приоритет	Оператор	Определение	**Примечания для неизвестных значений**	Принудительно преобразовать операнд к float [1]
1	**-**	Унарный минус	**-**Unknown → Unknown	Да
2	**not**	Логическое НЕ	**not** Unknown → Unknown	Да
3	*	Умножение	0 * Unknown → Unknown (да, Unknown, а не 0 — чтобы не терять Unknown в арифметических операциях) 1.2 * Unknown → Unknown	Да
	/	Деление	Unknown / 0 → ошибка Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Да
4	+	Арифметическое сложение	1.2 + Unknown → Unknown	Да
	-	Арифметическое вычитание	1.2 - Unknown → Unknown	Да
5	<	Меньше. Оператор определяется как: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	Да
	<=	Меньше или равно. Оператор определяется как: A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	Да
	>	Больше. Оператор определяется как: A>B ⇔ (A>B+0.000001)		Да

Приоритет	Оператор	Определение	**Примечания для неизвестных значений**	Принудительно преобразовать операнд к float [1]
	>=	Больше или равно. Оператор определяется как: A>=B ⇔ (A≥B-0.000001)		Да
6	**=**	Равно. Оператор определяется как: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		Нет [1]
	<>	Не равно. Оператор определяется как: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		Нет [1]
7	**and**	Логическое И	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	Да
8	**or**	Логическое ИЛИ	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	Да

[1] Строковый операнд по-прежнему преобразуется в числовой, если:

- другой операнд является числовым
- для операнда используется оператор, отличный от **=** или **<>**

(Если преобразование не удается, числовой операнд преобразуется в строковый, и оба операнда сравниваются как строки.)

Операторы **not**, **and** и **or** чувствительны к регистру и должны быть записаны строчными буквами. Они также должны быть отделены пробелами или скобками.

Все операторы, кроме унарных **-** и **not**, являются левоассоциативными. Унарные **-** и **not** не являются ассоциативными (то есть следует использовать **-(-1)** и **not (not 1)** вместо **--1** и **not not 1**).

Результат вычисления:

- операторы **<**, **<=**, **>**, **>=**, **=**, **<>** возвращают в выражении триггера ’1’, если указанное отношение истинно, и ’0’, если оно ложно. Если хотя бы один операнд Unknown, результат — Unknown;
- **and** для известных операндов возвращает ’1’, если оба операнда не равны ’0’; в противном случае возвращает ’0’; для неизвестных операндов **and** возвращает ’0’ только если один из операндов равен ’0’; в противном случае возвращает ’Unknown’;
- **or** для известных операндов возвращает ’1’, если хотя бы один из операндов не равен ’0’; в противном случае возвращает ’0’; для неизвестных операндов **or** возвращает ’1’ только если один из операндов не равен ’0’; в противном случае возвращает ’Unknown’;
- результат логического отрицания **not** для известного операнда — ’0’, если значение операнда не равно ’0’; ’1’, если значение операнда равно ’0’. Для неизвестного операнда **not** возвращает ’Unknown’.

Неизвестное состояние выражения

В выражении триггера может появиться неизвестный операнд, когда:

- используется неподдерживаемый элемент данных
- при вычислении функции для поддерживаемого элемента данных возникает ошибка

В этом случае выражение триггера обычно вычисляется как `Unknown` (так как его невозможно вычислить)

Можно получать уведомления о неизвестных триггерах.

Исключения

Несмотря на неизвестный операнд, выражения триггера в некоторых случаях могут вычисляться в известный результат (Problem/OK):

- `nodata()` функция вычисляется независимо от того, поддерживается ли указанный элемент данных или нет.
- Выражения с AND/OR могут быть вычислены в известный результат в двух случаях:
  - **Случай 1**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" вычисляется в известный результат ('1' или "Problem"),
  - **Случай 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" вычисляется в известный результат ('0' или "OK").
- Если вычисление функции для поддерживаемого элемента данных завершается ошибкой, значение функции становится `Unknown`, и оно участвует как неизвестный операнд в дальнейшем вычислении выражения.

Неизвестные операнды могут "исчезать" только в логических выражениях, как описано выше. В арифметических выражениях неизвестные операнды всегда приводят к `Unknown` (кроме деления на 0).

Attention:
Неизвестное состояние выражения не изменяет состояние триггера (Problem/OK). Поэтому, если состояние триггера было "Problem" (см. Случай 1), оно остается в состоянии проблемы, даже если известная часть разрешается ('1' становится '0'), потому что выражение теперь вычисляется как `Unknown`, а это не изменяет состояние триггера.

Если выражение триггера с несколькими неподдерживаемыми элементами данных вычисляется как `Unknown`, сообщение об ошибке во веб-интерфейсе относится к последнему вычисленному неподдерживаемому элементу данных.

Кэширование значений

Значения, которые необходимы для вычисления триггеров, кэшируются Zabbix сервером. По этой причине такое вычисление триггеров приводит к более высокой нагрузке на базу данных на некоторое время после перезапуска сервера. Кэш значений не очищается, когда значения истории элементов данных удаляются (либо вручную, либо при помощи автоматической очистки истории), поэтому сервер будет использовать кэшированные значения, пока они не станут старше, чем периоды времени, заданные в функциях триггеров, либо пока сервер не будет перезапущен.

Note:
Если в кэше нет последних данных и в функции не определён период запроса, Zabbix по умолчанию будет запрашивать у базы данных значения истории вплоть до одной недели назад.

Агрегатные функции

Если не указано иное, все перечисленные здесь функции поддерживаются в:

- выражениях триггеров
- формулах вычисляемых элементов данных
- макросах выражений

Агрегатные функции могут работать либо с:

- историей элементов данных, например, `min(/host/key,1h)`
- функциями foreach в качестве единственного параметра, например, `min(last_foreach(/*/key))` (только в формулах вычисляемых элементов данных; чтобы использовать в триггерах, укажите вычисляемый элемент данных по его ключу)

Функции перечислены без дополнительной информации. Щелкните функцию, чтобы увидеть полные сведения.

Function	Description
avg	Среднее значение элемента данных за заданный период вычисления.
bucket_percentile	Вычисляет процентиль по корзинам гистограммы.
count	Количество значений в массиве, возвращаемом функцией foreach.
histogram_quantile	Вычисляет φ-квантиль по корзинам гистограммы.

Function	Description
item_count	Количество существующих элементов данных в конфигурации, соответствующих критериям фильтра.
kurtosis	”Хвостатость” распределения вероятностей в собранных значениях за заданный период вычисления.
mad	Медианное абсолютное отклонение в собранных значениях за заданный период вычисления.
max	Наибольшее значение элемента данных за заданный период вычисления.
min	Наименьшее значение элемента данных за заданный период вычисления.
skewness	Асимметрия распределения вероятностей в собранных значениях за заданный период вычисления.
stddevpop	Стандартное отклонение генеральной совокупности в собранных значениях за заданный период вычисления.
stddevsamp	Выборочное стандартное отклонение в собранных значениях за заданный период вычисления.
sum	Сумма собранных значений за заданный период вычисления.
sumofsquares	Сумма квадратов в собранных значениях за заданный период вычисления.
varpop	Дисперсия генеральной совокупности собранных значений за заданный период вычисления.
varsamp	Выборочная дисперсия собранных значений за заданный период вычисления.

Общие параметры

- `/host/key` — общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети.
- `(sec|#num)<:time shift>` — общий второй параметр для функций, ссылающихся на историю элемента данных узла сети, где:
  - **sec** — максимальный период оценки в секундах (можно использовать суффиксы времени).
  - **#num** — максимальный диапазон оценки в последних собранных значениях (если перед параметром стоит знак решётки).
  - **time shift** (необязательный) позволяет переместить точку оценки назад во времени. Подробнее о задании временного сдвига см. в разделе Временной сдвиг.

Подробности функций

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются угловыми скобками < >
- Параметры / _ / и (|#)<: _ > никогда не должны заключаться в кавычки

avg(/host/key,(sec|#num)<:time shift>)

Среднее значение элемента данных в течение заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемые функции foreach: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Параметры: см. общие параметры.

Сдвиг времени полезен, когда нужно сравнить текущее среднее значение со средним значением некоторое время назад.

Примеры:

```
avg(/host/key,1h) #
avg(/host/key,#5) #
avg(/host/key,1h:now-1d) #                          25           24
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #                                  MyS
avg(/host/proc.num,5m)>300 #    ,                                5           300
```

bucket_percentile(item filter,time period,percentage)

Вычисляет процентиль по корзинам гистограммы.

Параметры:

- **item filter** - см. item filter;
- **time period** - см. time period;

- **percentage** - процент (0-100).

Комментарии:

- Поддерживается только в вычисляемых элементах данных;
- Эта функция является псевдонимом для `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Примеры:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #      95-                    5
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #      99-             API
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

Количество значений в массиве, возвращаемом функцией foreach.
 Поддерживаемые функции foreach: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Параметры:

- **func_foreach** - функция foreach, для которой нужно подсчитать количество возвращаемых значений. Подробности см. в разделе функции foreach. Обратите внимание, что count_foreach и bucket_rate_foreach поддерживают дополнительные параметры.
- **item filter** - см. синтаксис фильтра элементов данных;
- **time period** - см. период времени;
- **operator** (должен быть в двойных кавычках). Поддерживаемые `operators`:
eq - равно
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like - совпадает, если содержит шаблон (с учетом регистра)
bitand - побитовое И
regexp - совпадение с регулярным выражением, заданным в `pattern`, с учетом регистра
iregexp - совпадение с регулярным выражением, заданным в `pattern`, без учета регистра
- **pattern** - требуемый шаблон (строковые аргументы должны быть в двойных кавычках); поддерживается, если *operator* указан в третьем параметре.

Комментарии:

- Использование **count()** с функцией foreach, работающей с историческими данными (max_foreach, avg_foreach и т. д.), может повлиять на производительность, тогда как использование **exists_foreach()**, которая работает только с конфигурационными данными, такого эффекта не вызовет.
- Необязательные параметры *operator* или *pattern* нельзя оставлять пустыми после запятой, их можно только полностью опустить.
- При использовании *bitand* в качестве третьего параметра четвертый параметр `pattern` можно указать как два числа, разделенные символом '/': **number_to_compare_with/mask**. count() вычисляет "побитовое И" для значения и *mask* и сравнивает результат с *number_to_compare_with*. Если результат "побитового И" равен *number_to_compare_with*, значение учитывается.
Если *number_to_compare_with* и *mask* равны, достаточно указать только *mask* (без '/').
- При использовании *regexp* или *iregexp* в качестве третьего параметра четвертый параметр `pattern` может быть обычным или глобальным регулярным выражением (начинающимся с '@'). В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для сопоставления по regexp значения с плавающей точкой всегда будут представлены с 4 знаками после '.'. Также обратите внимание, что для больших чисел различие в десятичном (хранимом в базе данных) и двоичном (используемом сервером Zabbix) представлении может повлиять на 4-й десятичный знак.

Примеры:

```
count(max_foreach(/*/net.if.in[*],1h)) #                    net.if.in,
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #                    ,          95%
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Вычисляет φ-квантиль по корзинам гистограммы.
 Поддерживаемая функция foreach: *bucket_rate_foreach*.

Параметры:

- **quantile** - $0 \le \varphi \le 1$;
- **bucketN, valueN** - пары параметров, введенные вручную (>=2), или ответ bucket_rate_foreach.

Комментарии:

- Поддерживается только в вычисляемых элементах данных;
- Функционально соответствует 'histogram_quantile' в PromQL;
- Возвращает -1, если значения последней корзины 'Infinity' ("+*inf*") равны 0.

Примеры:

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

Количество существующих элементов данных в конфигурации, которые соответствуют критериям фильтра.
 Поддерживаемый тип значения: *Integer*.

Параметр:

- **item filter** - критерии выбора элементов данных, позволяют ссылаться по группе узлов сети, узлу сети, ключу элемента данных и тегам. Поддерживаются подстановочные знаки. См. item filter для получения дополнительных сведений.

Комментарии:

- Поддерживается только в вычисляемых элементах данных;
- Работает как псевдоним для функции *count(exists_foreach(item_filter))*.

Примеры:

```
item_count(/*/agent.ping?[group="Host group 1"]) #calculate the number of hosts with the agent.ping item i
```

kurtosis(/host/key,(sec|#num)<:time shift>)

”Хвостатость” распределения вероятностей в собранных значениях за заданный период оценки. См. также: Kurtosis.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Примеры:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigger on sharp spikes of CPU utilization over 5 minutes
```

mad(/host/key,(sec|#num)<:time shift>)

Медианное абсолютное отклонение в собранных значениях за заданный период оценки. См. также: Median absolute deviation.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Примеры:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

Наибольшее значение элемента данных в течение заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемые foreach functions: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Параметры: см. общие параметры.

Примеры:

```
max(/host/key,1h) - min(/host/key,1h) #                                                    (
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #                          Li
```

min(/host/key,(sec|#num)<:time shift>)

Наименьшее значение элемента данных в пределах заданного периода вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемые foreach functions: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Параметры: см. common parameters.

Примеры:

```
max(/host/key,1h) - min(/host/key,1h) #calculate the difference between the maximum and minimum values wit
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #return
```

skewness(/host/key,(sec|#num)<:сдвиг времени>)

Асимметрия распределения вероятностей в собранных значениях за заданный период оценки. См. также: Skewness.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Примеры:

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigger on sudden negative drops indicating crashes/restarts
```

stddevpop(/host/key,(sec|#num)<:time shift>)

Популяционное стандартное отклонение в собранных значениях за заданный период оценки. См. также: Standard deviation.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Примеры:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #trigger if CPU usage fluctuates heavily (population standar
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

Выборочное стандартное отклонение среди собранных значений за заданный период оценки. См. также: Standard deviation.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Для работы этой функции требуется как минимум два значения данных.

Примеры:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #     ,          CPU            (                    7    1
```

sum(/host/key,(sec|#num)<:time shift>)

Сумма собранных значений за определенный период вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемые foreach functions: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Параметры: см. common parameters.

Примеры:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #calculate the total incoming network traffic f
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #calculate the total disk space for all
sum(last_foreach(/*/net.dns[*,*,*])) #calculate the total number of successful DNS checks
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

Сумма квадратов собранных значений в пределах заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Примеры:

```
sumofsquares(/host/calculated.net.if.rate,1m) #                              (              )
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #
```

varpop(/host/key,(sec|#num)<:time shift>)

Дисперсия совокупности собранных значений в пределах заданного периода оценки. См. также: Variance.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Примеры:

```
varpop(/host/system.cpu.util[,user],10m)>50 #     ,          CPU            (          50    10     )
```

varsamp(/узел сети/key,(sec|#num)<:time shift>)

Выборочная дисперсия собранных значений в пределах заданного периода оценки. См. также: Variance.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая foreach function: *last_foreach*.

Параметры: см. common parameters.

Для работы этой функции требуется как минимум два значения данных.

Примеры:

```
varsamp(/     /system.cpu.util[,user],10m)>50 #trigger,          CPU           (                50   10
```

См. все поддерживаемые функции.

Функции foreach

Обзор

Функции foreach используются в агрегатных вычислениях для возврата одного агрегированного значения для каждого элемента данных, который выбран с помощью используемого **фильтра элементов данных**. Возвращается массив значений.

Например, функция `avg_foreach` вернет массив значений, где каждое значение — это *среднее* значение истории выбранного элемента данных за указанный временной интервал.

Attention:
Функции foreach поддерживаются только в формулах элемента данных Calculated. Их нельзя вызывать в обычных ключах элементов данных или использовать напрямую для построения выражений триггера. В выражениях триггера можно использовать элемент данных Calculated, обращаясь к его ключу.

Фильтр элементов данных является частью синтаксиса, используемого функциями foreach. В фильтре элементов данных поддерживается использование подстановочных знаков, поэтому необходимые элементы можно выбирать очень гибко.

Поддерживаемые функции

Function	Description
avg_foreach	Возвращает среднее значение для каждого элемента данных.
bucket_rate_foreach	Возвращает пары (верхняя граница корзины, значение скорости), пригодные для использования в функции `histogram_quantile()`, где "верхняя граница корзины" — это значение параметра ключа элемента данных, определенное параметром <parameter number>.
count_foreach	Возвращает количество значений для каждого элемента данных.
exists_foreach	Возвращает '1' для каждого включенного элемента данных.
last_foreach	Возвращает последнее значение для каждого элемента данных.
max_foreach	Возвращает максимальное значение для каждого элемента данных.
min_foreach	Возвращает минимальное значение для каждого элемента данных.
sum_foreach	Возвращает сумму значений для каждого элемента данных.

Синтаксис функций

Функции foreach поддерживают два общих параметра: _ _ (подробнее смотрите ниже) и _ :

```
   _foreach(   _    _   ,   _   )
```

Например:

`avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)`

вернёт среднее за пять минут значение для каждого из элементов данных «mysql.qps» в группе «MySQL server».

Обратите внимание, что некоторые функции поддерживают дополнительные параметры.

Синтаксис фильтра элементов данных

Фильтр элементов данных:

```
/  _  /  [    ]?[   ]
```

состоит из четырёх частей, где:
 _ — имя узла сети;
 — ключ элемента данных (без параметров);
 — параметры ключа элемента данных;
 — условия на основе группы узлов сети и/или тега элемента данных (как выражение).

Пробелы допустимы только внутри выражения условий.

Использование подстановочных символов

- Подстановочный символ может быть использован для замены имени узла сети, ключа элемента данных или отдельного параметра ключа элемента данных.
- Либо узел сети, либо ключ элемента данных должны быть указаны без подстановочного символа. Так, / _ /* и /*/ являются корректными фильтрами, но /*/* некорректен.
- Подстановочный символ не может быть использован как *часть* имени узла сети, ключа элемента данных, параметра ключа элемента данных.
- Подстановочный символ соответствует не более чем одному параметру ключа элемента данных. Поэтому подстановочный символ должен быть указан для каждого параметра отдельно (т.е. [abc,*,*]).

Выражение условий

Выражение условий поддерживает:

- Операнды:
 group — группа узлов сети;
 tag — тег элемента данных;
 "< >" — строковая константа, с использованием символа обратной косой черты \ для экранирования символов " и \.
- Операторы чувствительного к регистру сравнения строк: =, <>
- Логические операторы: and, or, not
- Группировка при помощи круглых скобок: ( )

Строковые константы обязательно должны заключаться в кавычки. Поддерживается только чувствительное к регистру сравнение полных строк.

Warning:
При указании в фильтре тегов (т.е. tag=" _ : "), двоеточие (":") используется как разделитель. Всё после него рассматривается как значение тега. Поэтому в данный момент не поддерживается указание тегов, содержащих двоеточие (":") внутри имени.

Примеры

Допустимы составные фильтры, ссылающиеся на ключ элемента данных, группу узлов сети и теги, как проиллюстрировано следующими примерами:

Пример синтаксиса	Описание
/ _ / [abc,*]	Соответствует подобным элементам данных на конкретном узле сети.
/*/	Соответствует одинаковому элементу данных на любом узле сети.
/*/ ?[group="ABC" and tag=" _ : "]	Соответствует тому же элементу данных любого узла сети из группы ABC, имеющему тег «имя_тега:значение».
/*/ [a,*,c]?[(group="ABC" and tag=" 1") or (group="DEF" and (tag=" 2" or tag=" 3: "))]	Соответствует подобным элементам данных любого узла сети из групп ABC или DEF с соответствующими тегами.

Все упомянутые элементы данных должны существовать и собирать данные. В вычисления включаются только активированные элементы данных на активированных узлах сети. Элементы данных в неподдерживаемом состоянии (в вычисления) не включаются.

Attention:
Если ключ упомянутого элемента данных меняется, фильтр должен быть обновлён вручную.

Упоминание родительской группы узлов сети включает родительскую и все вложенные группы узлов сети с их элементами данных.

Период времени

Параметр **second** позволяет указать период времени для агрегации. Период времени можно указывать только в единицах времени; количество значений (с префиксом **#**) не поддерживается.

Для удобства в этом параметре можно использовать поддерживаемые обозначения единиц измерения, например 5m (пять минут) вместо 300s (300 секунд) или 1d (один день) вместо 86400 (86400 секунд).

Для функции last_foreach период времени является необязательным параметром (поддерживается начиная с Zabbix 7.0), поэтому его можно не указывать:

```
last_foreach(/*/key?[group="host group"])
```

Период времени не поддерживается функцией `exists_foreach`.

Дополнительные параметры

bucket_rate_foreach

Функция `bucket_rate_foreach` поддерживает третий необязательный параметр:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

где `<parameter number>` — это позиция значения "bucket" в ключе элемента данных. Например, если значение "bucket" в `myItem[aaa,0.2]` равно `0.2`, то его позиция — 2.

Значение по умолчанию для <parameter number> — 1.

count_foreach

Функция `count_foreach` поддерживает третий и четвертый необязательные параметры:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Где:

- **operator** — условный оператор для значений элемента данных (должен быть заключен в двойные кавычки). Поддерживаемые `operators`:
eq - равно
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like - совпадает, если содержит шаблон (с учетом регистра)
bitand - побитовое И
regexp - совпадение с регулярным выражением, заданным в `pattern`, с учетом регистра
iregexp - совпадение с регулярным выражением, заданным в `pattern`, без учета регистра.
- **pattern** — требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки); поддерживается, если *operator* указан в третьем параметре.

Примечания:

- Необязательные параметры *operator* или *pattern* нельзя оставлять пустыми после запятой, их можно только полностью опустить.
- При использовании *bitand* в качестве третьего параметра четвертый параметр `pattern` можно указать как два числа, разделенные '/': **number_to_compare_with/mask**. `count_foreach()` вычисляет побитовое И от значения и *mask* и сравнивает результат с *number_to_compare_with*. Если результат побитового И равен *number_to_compare_with*, значение учитывается.
Если *number_to_compare_with* и *mask* равны, достаточно указать только *mask* (без '/').
- При использовании *regexp* или *iregexp* в качестве третьего параметра четвертый параметр `pattern` может быть обычным или глобальным регулярным выражением (начинающимся с '@'). В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для сопоставления по regexp значения с плавающей точкой всегда будут представлены с 4 знаками после '.'. Также обратите внимание, что для больших чисел разница в десятичном (хранимом в базе данных) и двоичном (используемом сервером Zabbix) представлении может повлиять на 4-й десятичный знак.

См. агрегатные вычисления для получения дополнительных сведений и примеров использования функций foreach.

Поведение, зависящее от доступности

Следующая таблица иллюстрирует, как каждая функция ведёт себя в случае ограниченной доступности узла сети / элемента данных и данных истории.

Функция	Узел сети деактивирован	Узел сети с данными недоступен	Узел сети без данных недоступен	Элемент данных деактивирован	Элемент данных не поддерживается	Ошибка извлечения данных (SQL)
avg_foreach	игнорировать	вернуть среднее	игнорировать	игнорировать	игнорировать	игнорировать
bucket_rate_foreach	игнорировать	вернуть bucket rate	игнорировать	игнорировать	игнорировать	игнорировать
count_foreach	игнорировать	вернуть количество	0	игнорировать	игнорировать	игнорировать
exists_foreach	игнорировать	1	1	игнорировать	1	n/a
last_foreach	игнорировать	вернуть последнее	игнорировать	игнорировать	игнорировать	игнорировать
max_foreach	игнорировать	вернуть максимум	игнорировать	игнорировать	игнорировать	игнорировать

Функция	Узел сети деактивирован	Узел сети с данными недоступен	Узел сети без данных недоступен	Элемент данных деактивирован	Элемент данных не поддерживается	Ошибка извлечения данных (SQL)
min_foreach	игнорировать	вернуть минимум	игнорировать	игнорировать	игнорировать	игнорировать
sum_foreach	игнорировать	вернуть сумму	игнорировать	игнорировать	игнорировать	игнорировать

Если для элемента данных указано *игнорировать*, к агрегированию ничего не добавляется.

Побитовые функции

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Функции перечислены без дополнительной информации. Щелкните по функции, чтобы увидеть полные сведения.

Function	Description
bitand	Значение побитового AND для значения элемента данных и маски.
bitlshift	Побитовый сдвиг влево значения элемента данных.
bitnot	Значение побитового NOT для значения элемента данных.
bitor	Значение побитового OR для значения элемента данных и маски.
bitrshift	Побитовый сдвиг вправо значения элемента данных.
bitxor	Значение побитового исключающего OR для значения элемента данных и маски.

Подробности функции

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми
- в качестве параметров допустимы выражения
- необязательные параметры функций (или части параметров) обозначаются угловыми скобками < >

bitand(значение,маска)

Результат операции «побитовое И» значения элемента данных и маски.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **значение** — проверяемое значение;
- **маска** (обязательно) — 64-битное целое без знака (0 - 18446744073709551615).

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100.

Примеры:

```
bitand(last(/  _  /  ),12)=8 or bitand(last(/host/key),12)=4 #                    ,
bitand(last(/  _  /  ),20)=16 #              ,       —
```

bitlshift(значение,сдвиг_битов)

Побитовый сдвиг значения элемента данных влево.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **значение** — проверяемое значение;
- **сдвиг_битов** (обязательно) — количество битов для сдвига.

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100.

bitnot(значение)

Результат операции «побитовое НЕ» значения элемента данных.
 Поддерживаемые типы значений: *Integer*.

Параметр:

- **значение** — проверяемое значение.

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100

bitor(значение,маска)

Результат операции «побитовое ИЛИ» значения элемента данных и маски.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **значение** — проверяемое значение;
- **маска** (mandatory) — 64-битное целое без знака (0 - 18446744073709551615).

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100.

bitrshift(значение,сдвиг_битов)

Побитовый сдвиг значения элемента данных вправо.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **значение** — проверяемое значение;
- **сдвиг_битов** (обязательно) — количество битов для сдвига.

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100.

bitxor(значение,маска)

Результат операции «побитовое исключающее ИЛИ» значения элемента данных и маски.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **значение** — проверяемое значение;
- **маска** (обязательно) — 64-битное целое без знака (0 - 18446744073709551615).

Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путём сравнения с числом 4, а не 100.

См. все поддерживаемые функции.

Функции даты и времени

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Attention:
Функции даты и времени нельзя использовать в выражении сами по себе; в выражение должна быть включена как минимум одна функция из другой группы, ссылающаяся на элемент данных узла сети, за исключением функции nodata(). Подробную информацию о том, как функции даты и времени работают в выражениях, см. в разделе Время вычисления.

Функции перечислены без дополнительной информации. Щелкните по функции, чтобы увидеть полные сведения.

Function	Description
date	Текущая дата в формате YYYYMMDD.
dayofmonth	День месяца в диапазоне от 1 до 31.
dayofweek	День недели в диапазоне от 1 до 7.
now	Количество секунд с начала эпохи (00:00:00 UTC, 1 января 1970 года).
time	Текущее время в формате HHMMSS.

Подробности функций

date

Текущая дата в формате ГГГГММДД (YYYYMMDD).

Пример:

`date()<20220101`

dayofmonth

Число месяца в диапазоне от 1 до 31.

Пример:

`dayofmonth()=1`

dayofweek

День недели в диапазоне от 1 до 7 (Пн — 1, Вс — 7).

Пример (только рабочие дни):

`dayofweek()<6`

Пример (только выходные):

`dayofweek()>5`

now

Количество секунд с начала эпохи Unix (00:00:00 1 января 1970 года по Гринвичу).

Пример:

`now()<1640998800`

time

Текущее время в формате ЧЧММСС (HHMMSS).

Пример (только ночное время, 00:00—06:00):

`time()<060000`

См. все поддерживаемые функции.

Функции истории

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы просмотреть подробные сведения.

Function	Description
change	Величина разницы между предыдущим и последним значением.
changecount	Количество изменений между соседними значениями в течение заданного периода оценки.
count	Количество значений в течение заданного периода оценки.
countunique	Количество уникальных значений в течение заданного периода оценки.
find	Поиск совпадения значения в течение заданного периода оценки.
first	Первое (самое старое) значение в течение заданного периода оценки.
firstclock	Метка времени первого (самого старого) значения в течение заданного периода оценки.
fuzzytime	Проверка расхождения времени пассивного агента со временем сервера/прокси Zabbix.
jsonpath	Возвращает массив результатов запроса JSONPath.
last	Самое последнее значение.
lastclock	Метка времени N-го по давности значения в течение заданного периода оценки.
logeventid	Проверка соответствия идентификатора события последней записи журнала регулярному выражению.
logseverity	Уровень серьезности последней записи журнала.
logsource	Проверка соответствия источника последней записи журнала регулярному выражению.
logtimestamp	Метка времени сообщения журнала для N-го по давности значения элемента данных журнала.

Function	Description
monodec	Проверка наличия монотонного уменьшения значений.
monoinc	Проверка наличия монотонного увеличения значений.
nodata	Проверка отсутствия полученных данных.
percentile	P-й процентиль периода, где P (процент) указывается третьим параметром.
rate	Средняя скорость увеличения в секунду монотонно возрастающего счетчика в течение заданного периода времени.
xmlxpath	Возвращает массив результатов запроса XML XPath.

Общие параметры

- `/host/key` — общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети.
- `(sec|#num)<:time shift>` — общий второй параметр для функций, ссылающихся на историю элемента данных узла сети, где:
  - **sec** — максимальный период вычисления в секундах (можно использовать суффиксы времени), или
  - **#num** — максимальный диапазон вычисления в последних собранных значениях (если перед ним стоит знак решётки).
  - **time shift** (необязательный) позволяет переместить точку вычисления назад во времени. Подробнее о задании временного сдвига см. в разделе подробнее.

Подробности функций

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми
- необязательные параметры функций (или части параметров) обозначаются угловыми скобками < >
- параметры, специфичные для функций, описываются с каждой функцией
- параметры / _ / и (|#)<: _ > никогда не должны заключаться в кавычки

change(/host/key)

Величина разницы между предыдущим и последним значением.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.
 Для строк возвращает: 0 - значения равны; 1 - значения различаются.

Параметры: см. общие параметры.

Комментарии:

- Числовая разница будет вычислена, как показано на следующих примерах входных значений ('previous' и 'latest' value = разница):
'1' и '5' = +4
'3' и '1' = −2
'0' и '-2.5' = −2.5.
- В триггерах полезно для обнаружения резких всплесков (или падений), сбросов счетчиков, числовых переходов состояния.
- См. также: abs для сравнения.

Примеры:

```
change(/host/system.uptime)<0 #                                          (                )
change(/host/system.cpu.load[all,avg1])>2 #          (         )              2
change(/host/vfs.fs.size[/,free])<-1G #                                  1
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

Количество изменений между соседними значениями в пределах заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **mode** (должен быть в двойных кавычках) - возможные значения: *all* - считать все изменения (по умолчанию); *dec* - считать уменьшения; *inc* - считать увеличения

Комментарии:

- Для нечисловых типов значений параметр *mode* игнорируется.
- В триггерах полезно для выявления частых изменений состояния (что указывает на нестабильность).

Примеры:

```
changecount(/host/icmpping,10m)>5 #      ping           5      10
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5 #      operstate       eth0
changecount(/host/proc.num[httpd],15m)>10 #            httpd           10      15
changecount(/host/key,#10,"inc") #                  (                 )          10
changecount(/host/key,24h,"dec") #                  (                 )         24
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Количество значений в пределах заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **operator** (должен быть заключен в двойные кавычки) Поддерживаемые operators:
eq - равно (по умолчанию для integer, float)
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like (по умолчанию для string, text, log) - совпадает, если содержит шаблон (с учетом регистра)
bitand - побитовое И
regexp - совпадение с регулярным выражением, заданным в pattern, с учетом регистра
iregexp - совпадение с регулярным выражением, заданным в pattern, без учета регистра
- **pattern** - требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки)

Комментарии:

- Элементы данных типа Float сопоставляются с точностью 2.22e-16.
- *like* не поддерживается как оператор для значений integer.
- *like* и *bitand* не поддерживаются как операторы для значений float.
- Для значений string, text и log поддерживаются только операторы *eq*, *ne*, *like*, *regexp* и *iregexp*.
- При использовании *bitand* в качестве оператора четвертый параметр pattern можно указать как два числа, разделенные символом '/': **number_to_compare_with/mask**. count() вычисляет "побитовое И" от значения и *mask* и сравнивает результат с *number_to_compare_with*. Если результат "побитового И" равен *number_to_compare_with*, значение учитывается.
Если *number_to_compare_with* и *mask* равны, достаточно указать только *mask* (без '/').
- При использовании *regexp* или *iregexp* в качестве оператора четвертый параметр pattern может быть обычным или глобальным регулярным выражением (начинающимся с '@'). В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для сопоставления regexp значения float всегда будут представлены с 4 десятичными знаками после '.'. Также обратите внимание, что для больших чисел разница в десятичном (хранимом в базе данных) и двоичном (используемом сервером Zabbix) представлении может повлиять на 4-ю десятичную цифру.

Примеры:

```
count(/host/icmpping,30m,,"0")>5 #ping          5      30
count(/host/key,10m,"like","error") #                  10                ,        'error'
count(/host/key,10m,,12) #                   10                ,      '12'
count(/host/key,10m,"gt",12) #                     10                ,         '12'
count(/host/key,#10,"gt",12) #                    10                 ,         '12'
count(/host/key,10m:now-1d,"gt",12) #               24       10        24                    ,
count(/host/key,10m,"bitand","6/7") #                  10                ,      '110' (         )   3
count(/host/key,10m:now-1d) #               24       10        24
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Количество уникальных значений за заданный период оценки.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **operator** (должен быть заключен в двойные кавычки). Поддерживаемые operators:
eq - равно (по умолчанию для integer, float)
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like (по умолчанию для string, text, log) - совпадает, если содержит шаблон (с учетом регистра)
bitand - побитовое И
regexp - совпадение с регулярным выражением, заданным в pattern, с учетом регистра
iregexp - совпадение с регулярным выражением, заданным в pattern, без учета регистра
- **pattern** - требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки)

Комментарии:

- Элементы данных типа Float сопоставляются с точностью 2.22e-16.
- *like* не поддерживается как оператор для значений integer.
- *like* и *bitand* не поддерживаются как операторы для значений float.

- Для значений string, text и log поддерживаются только операторы *eq*, *ne*, *like*, *regexp* и *iregexp*.
- При использовании *bitand* в качестве оператора четвертый параметр `pattern` можно указать как два числа, разделенные '/': **number_to_compare_with/mask**. countunique() вычисляет "побитовое И" для значения и *mask* и сравнивает результат с *number_to_compare_with*. Если результат "побитового И" равен *number_to_compare_with*, значение учитывается.
Если *number_to_compare_with* и *mask* равны, достаточно указать только *mask* (без '/').
- При использовании *regexp* или *iregexp* в качестве оператора четвертый параметр `pattern` может быть обычным или глобальным регулярным выражением (начинающимся с '@'). В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для сопоставления regexp значения float всегда будут представлены с 4 десятичными знаками после '.'. Также обратите внимание, что для больших чисел разница в десятичном (хранимом в базе данных) и двоичном (используемом сервером Zabbix) представлении может повлиять на 4-й десятичный знак.

Примеры:

```
countunique(/host/key,10m) #                          10
countunique(/host/key,10m,"like","error") #                        10                ,           'error'
countunique(/host/key,10m,,12) #                        10                ,      '12'
countunique(/host/key,10m,"gt",12) #                          10                ,           '12'
countunique(/host/key,#10,"gt",12) #                            10                  ,           '12'
countunique(/host/key,10m:now-1d,"gt",12) #                       24        10         24                        ,
countunique(/host/key,10m,"bitand","6/7") #                           10                ,      '110' (
countunique(/host/key,10m:now-1d) #                         24        10         24
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Находит совпадение значения в пределах заданного периода оценки.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.
 Возвращает: 1 - найдено; 0 - в противном случае.

Параметры:

- См. общие параметры
- **sec** или **#num** (необязательно) - по умолчанию используется последнее значение, если не указано
- **operator** (должен быть в двойных кавычках) Поддерживаемые `operators`:
eq - равно (по умолчанию для integer, float)
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like (по умолчанию для string, text, log) - совпадает, если содержит строку, указанную в `pattern` (с учетом регистра)
bitand - побитовое И
regexp - совпадение с регулярным выражением, чувствительное к регистру, указанным в `pattern`
iregexp - совпадение с регулярным выражением, нечувствительное к регистру, указанным в `pattern`
- **pattern** - требуемый шаблон (строковые аргументы должны быть в двойных кавычках); регулярное выражение Perl Compatible Regular Expression (PCRE), если `operator` - *regexp*, *iregexp*

Комментарии:

- Если обрабатывается более одного значения, возвращается '1', если найдено хотя бы одно совпадающее значение.
- *like* не поддерживается как оператор для значений integer.
- *like* и *bitand* не поддерживаются как операторы для значений float.
- Для значений string, text и log поддерживаются только операторы *eq*, *ne*, *like*, *regexp* и *iregexp*.
- При использовании *regexp* или *iregexp* в качестве оператора четвертый параметр `pattern` может быть обычным или глобальным регулярным выражением (начинающимся с '@'). В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения.

Примеры:

```
find(/host/key,10m,"like","error") #          ,        'error',          10
find(/host/agent.version,,"like","beta")=1 #    Zabbix     beta-    ,
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #                  -
```

first(/узел_сети/ключ,сек<:сдвиг_времени>)

Первое (самое старое) значение за указанный период вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- смотрите общие параметры

Смотрите также last().

Пример:

```
first(/  _  /  ,1h) #
```

firstclock(/host/key,sec<:time shift>)

Метка времени самого старого значения в пределах заданного периода вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры.

Функция завершается с ошибкой, если за указанный период не было собрано данных.

См. также lastclock().

Примеры:

```
firstclock(/host/key,1h) #
firstclock(/host/key,1h:now-24h) #
```

fuzzytime(/host/key,sec)

Проверяет, насколько время пассивного агента отличается от времени сервера/прокси Zabbix.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Возвращает: 1 - разница между значением элемента данных пассивного опроса (как временной меткой) и временной меткой сервера/прокси Zabbix (временем сбора значения) меньше или равна *sec* секундам; 0 - в противном случае.

Параметры:

- См. общие параметры.

Комментарии:

- Обычно используется с элементом данных 'system.localtime' для проверки того, что локальное время синхронизировано с локальным временем сервера Zabbix. *Обратите внимание*, что 'system.localtime' должен быть настроен как пассивная проверка для агента Zabbix; в Zabbix agent 2 он может быть настроен как активная проверка.
- Также может использоваться с ключом `vfs.file.time[/path/file,modify]` для проверки того, что файл долгое время не обновлялся.
- Не рекомендуется использовать эту функцию в сложных выражениях триггеров (с участием нескольких элементов данных), поскольку это может привести к неожиданным результатам (разница во времени будет измеряться по самой последней метрике), например, в `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Примеры:

```
fuzzytime(/host/system.localtime,5m)=0 #                                /    Zabbix         5
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Возвращает массив результатов запроса JSONPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **pattern** — обязательный шаблон (строковые аргументы необходимо заключать в двойные кавычки)

Комментарии:

- Эта функция используется с агрегатными функциями.
- Запрос JSONPath выполняется для каждого выбранного значения истории; каждый непустой результат успешного запроса добавляется в массив. Если результат запроса для одного значения истории содержит несколько совпадений, каждое совпадение добавляется в массив отдельно.
- Функция завершается с ошибкой, если:
  - Формат `pattern` недействителен.
  - Любое из значений истории содержит недействительный JSON или запрос JSONPath завершается ошибкой для этого значения.
- Функция возвращает пустой массив, если:
  - Значения истории не выбраны.
  - Все выбранные значения истории содержат недействительный JSON.

Пример:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

last(/host/key,<#num<:time shift»)

Самое последнее значение.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **#num** (необязательно) - N-е по счету самое последнее значение

Комментарии:

- Обратите внимание, что период времени с хэштегом (#N) здесь работает иначе, чем во многих других функциях. Например: `last(/host/key)` всегда равно `last(/host/key,#1)`; `last(/host/key,#3)` - третье по счету самое последнее значение (*а не* три последних значения).
- Zabbix не гарантирует точный порядок значений, если в истории за одну секунду существует более двух значений.
- См. также first().

Примеры:

```
last(/host/key) #
last(/host/key,#2) #
last(/host/key,#1)<>last(/host/key,#2) #
```

lastclock(/host/key,<#num<:time shift»)

Временная метка N-го по новизне значения в пределах заданного периода вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры;
- **#num** (необязательно) — N-е по новизне значение.

Функция завершается с ошибкой, если за указанный период не было собрано данных или не было собрано N-е значение.

См. также firstclock().

Примеры:

```
lastclock(/host/key) #
lastclock(/host/key,#2) #
now()-lastclock(/host/heartbeat.item)>300 #                          heartbeat         5
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #
```

logeventid(/узел_сети/ключ,<#число<:сдвиг_времени»,<шаблон>)

Проверка, соответствует ли указанному регулярному выражению идентификатор события последней записи из журнала.
 Поддерживаемые типы значений: *Log*.
 Возвращает: 0 — нет соответствия; 1 — соответствует.

Параметры:

- Смотрите общие параметры;
- **#число** (опционально) — N-ое из наиболее недавних значений;
- **шаблон** (опционально) — регулярное выражение, описывающее нужный шаблон, в формате Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки).

Примеры:

```
logeventid(/   _   /eventlog[Security],,"^4625$")=1 #                          ,          "4625" (
logeventid(/   _   /eventlog[System],,"^(6008|41)$")=1 #                         ,          "6008"    "41"
```

logseverity(/узел_сети/ключ,<#число<:сдвиг_времени»)

Важность события последней записи в журнале.
 Поддерживаемые типы значений: *Log*.
 Возвращает: 0 — важность по умолчанию; N — важность (целое число, полезно для журналов событий Windows: 1 — Уведомление, 2 — Предупреждение, 4 — Ошибка, 7 — Аудит отказов, 8 — Аудит успехов, 9 — Критическая ошибка, 10 — Детали).

Параметры:

- Смотрите общие параметры;
- **#число** (опционально) — N-ое из наиболее недавних значений.

Zabbix берёт важность журнала из поля **Информация** журнала событий Windows.

Примеры:

```
logseverity(/  _  /log[/var/log/syslog],10m)>3 #                              «3»
logseverity(/  _  /eventlog[System],10m)=4 #                    ,     «   »
```

logsource(/узел_сети/ключ,<#число<:сдвиг_времени»,<шаблон>)

Проверка, соответствует ли регулярному выражению источник последней записи в журнале.
 Поддерживаемые типы значений: *Log*.
 Возвращает: 0 — нет соответствия; 1 — соответствует.

Параметры:

- Смотрите общие параметры;
- **#число** (опционально) — N-ое из наиболее недавних значений;
- **шаблон** (опционально) — регулярное выражение, описывающее нужный шаблон, в формате Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки).

Обычно используется для журналов событий Windows.

Примеры:

```
logsource(/  _  /eventlog[Application],,"MSSQLSERVER")=1 #                    ,         «MSSQLSERVER»
logsource(/  _  /eventlog[System],,"Service Control Manager")=1 #                    ,          «Service
logsource(/  _  /eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^7
```

logtimestamp(/host/key,<#num<:time shift»)

Временная метка сообщения журнала для N-го самого последнего значения элемента данных журнала.
 Поддерживаемые типы значений: *Log*.

Параметры:

- См. общие параметры;
- **#num** (необязательно) — N-е самое последнее значение.

Комментарии:

- Вычисление сдвига времени основано на времени элемента данных, а не на временной метке сообщения журнала;
- Функция завершается с ошибкой, если:
  - получен элемент данных не типа журнала;
  - за указанный период не было собрано данных;
  - N-е значение не было собрано;
  - сообщение журнала не содержит значения временной метки.

Примеры:

```
logtimestamp(/host/key) #retrieve the timestamp of the latest log message
logtimestamp(/host/key,#2) #retrieve the timestamp of the 2nd latest log message
logtimestamp(/host/key,#3:now-1d) #retrieve the timestamp of the 3rd latest log message a day ago
```

monodec(/узел_сети/ключ,(сек|#число)<:сдвиг_времени>,<режим>)

Проверка наличия монотонного убывания значений.
 Поддерживаемые типы значений: *Integer*.
 Возвращает: 1, если все элементы в течение указанного периода времени непрерывно уменьшаются; 0 — в противном случае.

Параметры:

- Смотрите общие параметры
- **режим** (должен быть в двойных кавычках) — *weak* (каждое значение меньше или равно предыдущему; по умолчанию) или *strict* (каждое значение строго уменьшается)

Примеры:

```
monodec(/  _  /system.swap.size[all,free],60s) + monodec(/  _  2/system.swap.size[all,free],60s) + monode
monodec(/  _  /proc.num[nginx],10m,"strict")=1 #            nginx                      10
```

monoinc(/узел_сети/ключ,(сек|#число)<:сдвиг_времени>,<режим>)

Проверка наличия монотонного возрастания значений.
 Поддерживаемые типы значений: *Integer*.
 Возвращает: 1, если все элементы в течение указанного периода времени непрерывно возрастают; 0 — в противном случае.

Параметры:

- Смотрите общие параметры
- **режим** (должен быть в двойных кавычках) — *weak* (каждое значение больше или равно предыдущему; по умолчанию) или *strict* (каждое значение строго возрастает)

Примеры:

```
monoinc(/  _  /system.localtime,#3,"strict")=0 #
monoinc(/  _  /vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #      ,                    24    (
```

nodata(/host/key,sec,<mode>)

Проверка отсутствия полученных данных.
 Поддерживаемые типы значений: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Возвращает: 1 - если данные не получены в течение заданного периода времени; 0 - в противном случае.

Параметры:

- См. общие параметры
- **sec** - период не должен быть меньше 30 секунд, поскольку процесс синхронизации истории вычисляет эту функцию только каждые 30 секунд; `nodata(/host/key,0)` запрещено
- **mode** - если задано значение *strict* (в двойных кавычках), эта функция не будет зависеть от доступности прокси (подробности см. в комментариях)

Комментарии:

- По умолчанию триггеры 'nodata', отслеживаемые прокси, зависят от доступности прокси - если прокси становится недоступным, триггеры 'nodata' не сработают сразу после восстановления соединения, а пропустят данные за период задержки. Обратите внимание, что для пассивных прокси подавление активируется, если соединение восстановлено более чем через 15 секунд и не менее чем через 2 секунды. Для активных прокси подавление активируется, если соединение восстановлено более чем через 15 секунд. Чтобы отключить зависимость от доступности прокси, используйте третий параметр, например: `nodata(/host/key,5m,"strict")`; в этом случае функция сработает сразу после истечения периода оценки (пяти минут) без данных.
- Эта функция отобразит ошибку, если в течение периода, заданного первым параметром:
- отсутствуют данные и сервер Zabbix был перезапущен
- отсутствуют данные и обслуживание завершилось
- отсутствуют данные и элемент данных был добавлен или повторно включен.
- Ошибки отображаются в столбце *Информация* списка триггеров.
- Эта функция может работать некорректно, если время на сервере Zabbix, прокси и агенте различается. См. также: Требование синхронизации времени.

Пример:

```
nodata(/host/agent.ping,5m)=1 #          ,              Zabbix              5
```

percentile(/узел_сети/ключ,(сек|#число)<:сдвиг_времени>,процент)

P-й процентиль периода, где P (процент) указывается третьим параметром.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- Смотрите общие параметры;
- **процент** — число с плавающей точкой от 0 до 100 (включительно), до 4 цифр после десятичной точки.

Примеры:

```
percentile(/  _  /net.if.in[eth0,bytes],1h,95)>1000000 #95-                    (  /  )   1
percentile(/  _  /system.cpu.util,5m,95)>80 #95-                  CPU                  80%
percentile(/  _  /icmppingsec[192.168.0.2],15m,95)>0.15 #                              150  ,
percentile(/  _  /net.if.in[eth0,bytes],1h,50) #     50-       (          )                  ;
(percentile(/  _  /net.if.in[eth0,bytes],1h,50)+percentile(/  _  /net.if.in[eth0,bytes],1h,51))/2 #
```

rate(/host/key,sec<:time shift>)

Среднее почасовое увеличение монотонно возрастающего счетчика за заданный период времени.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры

Функционально соответствует 'rate' в PromQL.

Примеры:

```
rate(/host/key,30s) #              30          20,              0.67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #                    eth0      500  /        5
rate(/host/app.requests.count,1m)>100 #                          100
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Возвращает массив результатов запросов XML XPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **pattern** — обязательный шаблон (строковые аргументы необходимо заключать в двойные кавычки)

Комментарии:

- Эта функция используется с агрегатными функциями.
- Запрос XML XPath выполняется для каждого выбранного значения истории; каждый непустой успешный результат запроса добавляется в массив. Если результат запроса для одного значения истории содержит несколько совпадений, каждое совпадение добавляется в массив отдельно.
- Функция завершается с ошибкой, если:
  - Формат `pattern` недействителен.
  - Zabbix скомпилирован без поддержки XML.
  - Любое из значений истории содержит недействительный XML или запрос XPath завершается с ошибкой для этого значения.
- Функция возвращает пустой массив, если:
  - Значения истории не выбраны.
  - Все выбранные значения истории содержат недействительный XML.

Пример:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

См. все поддерживаемые функции.

Функции трендов

Функции трендов, в отличие от функций истории, используют данные трендов для вычислений.

Тренды хранят агрегированные значения по часам. Функции трендов используют эти почасовые средние, поэтому полезны для долгосрочного анализа.

Результаты функций трендов кэшируются, поэтому несколько вызовов одной и той же функции с одинаковыми параметрами получают данные из базы только один раз. Кэш функций трендов управляется параметром сервера TrendFunctionCacheSize.

Используйте `zabbix[tcache,cache,paccessed]`, чтобы оценить, какая часть кэша фактически используется. Низкие значения `paccessed` вместе с низкими `pitems` и `pmisses` обычно означают, что размер кэша можно уменьшить.

Триггеры, которые ссылаются **только** на функции трендов, вычисляются один раз за наименьший временной период в выражении. Например, триггер вида

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

будет вычисляться один раз в день. Если триггер содержит как функции трендов, так и функций истории (или даты и времени и/или nodata()), он вычисляется в соответствии с обычными принципами.

Все перечисленные здесь функции поддерживаются в:

- выражениях триггеров
- вычисляемых элементах данных
- макросах выражений

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы увидеть полные сведения.

Function	Description
baselinedev	Возвращает число отклонений (по алгоритму stddevpop) между последним периодом данных и теми же периодами данных в предыдущих сезонах.
baselinewma	Вычисляет базовую линию, усредняя данные за тот же временной интервал в нескольких равных временных периодах (”сезонах”) с использованием алгоритма взвешенного скользящего среднего.
trendavg	Среднее значение трендов за заданный период времени.
trendcount	Количество успешно полученных значений истории, использованных для вычисления значения тренда за заданный период времени.
trendmax	Максимальное значение трендов за заданный период времени.

Function	Description
trendmin	Минимальное значение трендов за заданный период времени.
trendstl	Возвращает уровень аномалий за период обнаружения - десятичное значение от 0 до 1, равное (( )/( )).
trendsum	Сумма значений трендов за заданный период времени.

Общие параметры

- / _ / является общим обязательным первым параметром
- _ : _ является общим вторым параметром, где:
  - **период_времени** — период времени (минимум «1h»), определённый как <N><единица_времени>, где N — количество единиц времени, _ — h (час), d (сутки), w (неделя), M (месяц) или y (год).
  - **сдвиг_времени** — смещение периода времени (смотрите примеры функций)

Подробности функций

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми
- необязательные параметры функций (или части параметров) обозначаются угловыми скобками < >
- параметры, специфичные для функций, описываются с каждой функцией
- параметры / _ / и _ : _ никогда не должны заключаться в кавычки

baselinedev(/узел_сети/ключ,период_данных:сдвиг_времени,сезон_ед,сезон_кол-во)

Возвращает количество стандартных отклонений (по алгоритму stddevpop, среднеквадратичное отклонение) между последним периодом данных и теми же периодами данных в предыдущие сезоны.

Параметры:

- Смотрите общие параметры
- **период_данных** — период сбора данных в течение сезона, определённый как <N><единица_времени>, где
N — количество единиц времени
 _ — h (час), d (сутки), w (неделя), M (месяц) или y (год), должно быть меньше или равно сезону
- **сезон_ед** — календарная единица, определяющая один сезон (h, d, w, M, y); не может быть меньше периода данных
- **сезон_кол-во** — количество оцениваемых сезонов

Примеры:

```
baselinedev(/  _  /  ,1d:now/d,"M",6) #                    (          )
baselinedev(/  _  /  ,1h:now/h,"d",10)  #
```

baselinewma(/узел_сети/ключ,период_данных:сдвиг_времени,сезон_ед,сезон_кол-во)

Вычисляет базовый уровень путем усреднения данных за один и тот же интервал времени в течение нескольких равных периодов времени («сезонов») с использованием алгоритма взвешенного скользящего среднего.

Параметры:

- Смотрите общие параметры
- **период_данных** — период сбора данных в течение сезона, определённый как <N><единица_времени>, где:
N — количество единиц времени
 _ — h (час), d (сутки), w (неделя), M (месяц) или y (год), должно быть меньше или равно сезону
Сдвиг_времени] — смещение периода времени, определяет конец интервала сбора данных в сезонах (смотрите примеры)
- **сезон_ед** — календарная единица, определяющая однин сезон (h, d, w, M, y); не может быть меньше периода данных
- **сезон_кол-во** — количество оцениваемых сезонов

Примеры:

```
baselinewma(/  _  /  ,1h:now/h,"d",3) #                                    ,          .
baselinewma(/  _  /  ,2h:now/h,"d",3) #                                              ,
baselinewma(/  _  /  ,1d:now/d,"M",4) #                    ,    «  »,      4    ,
```

trendavg(/узел_сети/ключ,период_времени:сдвиг_времени)

Среднее из значений динамики изменений за определённый период времени.

Параметры:

- Смотрите общие параметры

Примеры:

```
trendavg(/  _  /  ,1h:now/h) #                (     , 12:00-13:00)
trendavg(/  _  /  ,1h:now/h-1h) #               (11:00-12:00)
trendavg(/  _  /  ,1h:now/h-2h) #               (10:00-11:00)
trendavg(/  _  /  ,1M:now/M-1y) #
```

trendcount(/узел_сети/ключ,период_времени:сдвиг_времени)

Количество успешно извлечённых значений истории, используемых для расчёта значений динамики изменений за определённый период времени.

Параметры:

- Смотрите общие параметры

Примеры:

```
trendcount(/  _  /  ,1h:now/h) #                (     , 12:00-13:00)
trendcount(/  _  /  ,1h:now/h-1h) #               (11:00-12:00)
trendcount(/  _  /  ,1h:now/h-2h) #               (10:00-11:00)
trendcount(/  _  /  ,1M:now/M-1y) #
```

trendmax(/узел_сети/ключ,период_времени:сдвиг_времени)

Максимальное значение из динамики изменений за определённый период времени.

Параметры:

- Смотрите общие параметры

Примеры:

```
trendmax(/  _  /  ,1h:now/h) #                (     , 12:00-13:00)
trendmax(/  _  /  ,1h:now/h) - trendmin(/  _  /  ,1h:now/h) →                                (
trendmax(/  _  /  ,1h:now/h-1h) #                (11:00-12:00)
trendmax(/  _  /  ,1h:now/h-2h) #                (10:00-11:00)
trendmax(/  _  /  ,1M:now/M-1y) #
```

trendmin(/узел_сети/ключ,период_времени:сдвиг_времени)

Минимальное значение из динамики изменений за определённый период времени.

Параметры:

- Смотрите общие параметры

Примеры:

```
trendmin(/  _  /  ,1h:now/h) #                (     , 12:00-13:00)
trendmax(/  _  /  ,1h:now/h) - trendmin(/  _  /  ,1h:now/h) →                                (
trendmin(/  _  /  ,1h:now/h-1h) #                (11:00-12:00)
trendmin(/  _  /  ,1h:now/h-2h) #                (10:00-11:00)
trendmin(/  _  /  ,1M:now/M-1y) #
```

trendstl(/узел_сети/ключ,период_оценки:сдвиг_времени,период_обнаружения,сезон,<отклонения>,<devalg>,<s_window>)

Возвращает частоту аномалий за период обнаружения — десятичное значение от 0 до 1, которое определяется как (( )/( )).

Параметры:

- Смотрите общие параметры
- **период_оценки** — период времени, который необходимо разложить (минимум «1h»), определённый как <N><единица_времени>, где
N — количество единиц времени
 _ — h (час), d (сутки), w (неделя), M (месяц) или y (год)
- **период_обнаружения** — период времени до конца периода оценки, для которого вычисляются аномалии (минимум «1h», не может быть дольше периода оценки), определённый как <N><единица_времени>, где
N — количество единиц времени
 _ — h (час), d (сутки), w (неделя)
- **сезон** — самый короткий период времени, где ожидается повторяющийся шаблон («сезон»), (минимум «2h», не может быть дольше периода оценки, количество записей в периоде оценки должно быть больше двукратной

результирующей частоты (сезон/h)), определённой как <N><единица_времени>, где
N — количество единиц времени
  _ — h (час), d (сутки), w (неделя)

- **отклонения** — количество отклонений (вычисленных по алгоритму devalg), которые следует считать аномалией (может быть десятичным числом), (должно быть больше либо равно 1, по умолчанию: 3);
- **devalg** (должно быть в двойных кавычках) — алгоритм отклонений, может быть *stddevpop*, *stddevsamp* или *mad* (по умолчанию)
- **s_window** — протяжённость (в лагах) окна LOESS для извлечения сезонов (по умолчанию: 10 * количество записей в периоде оценки + 1)

Примеры:

```
trendstl(/  _  /  ,100h:now/h,10h,2h) #              100                  ,                      10
trendstl(/  _  /  ,100h:now/h-10h,100h,2h,2.1,"mad") #            100                  ,        10
trendstl(/  _  /  ,100d:now/d-1d,10d,1d,4,,10) #            100                              ,
trendstl(/  _  /  ,1M:now/M-1y,1d,2h,,"stddevsamp") #                        ,
```

trendsum(/узел_сети/ключ,период_времени:сдвиг_времени)

Сумма значений динамики изменений за определённый период времени.

Параметры:

- Смотрите общие параметры

Примеры:

```
trendsum(/  _  /  ,1h:now/h) #               (     , 12:00-13:00)
trendsum(/  _  /  ,1h:now/h-1h) #                   (11:00-12:00)
trendsum(/  _  /  ,1h:now/h-2h) #                   (10:00-11:00)
trendsum(/  _  /  ,1M:now/M-1y) #
```

См. все поддерживаемые функции.

Математические функции

Все функции, перечисленные здесь, поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Математические функции поддерживаются для типов значений float и integer, если не указано иное.

Функции перечислены без дополнительной информации. Щелкните по функции, чтобы увидеть полные сведения.

Function	Description
abs	Абсолютное значение значения.
acos	Арккосинус значения как угла, выраженного в радианах.
asin	Арксинус значения как угла, выраженного в радианах.
atan	Арктангенс значения как угла, выраженного в радианах.
atan2	Арктангенс координат ординаты (значения) и абсциссы, заданных как угол, выраженный в радианах.
avg	Среднее значение значений указанных элементов данных.
cbrt	Кубический корень значения.
ceil	Округлить значение вверх до ближайшего большего или равного целого числа.
cos	Косинус значения, где значение — это угол, выраженный в радианах.
cosh	Гиперболический косинус значения.
cot	Котангенс значения, где значение — это угол, выраженный в радианах.
degrees	Преобразует значение из радиан в градусы.
e	Число Эйлера (2.718281828459045).
exp	Число Эйлера в степени значения.
expm1	Число Эйлера в степени значения минус 1.
floor	Округлить значение вниз до ближайшего меньшего или равного целого числа.
log	Натуральный логарифм.
log10	Десятичный логарифм.
max	Наибольшее значение значений указанных элементов данных.
min	Наименьшее значение значений указанных элементов данных.

Function	Description
mod	Остаток от деления.
pi	Константа Pi (3.14159265358979).
power	Возведение значения в степень.
radians	Преобразует значение из градусов в радианы.
rand	Возвращает случайное целое значение.
round	Округлить значение до указанного числа знаков после запятой.
signum	Возвращает '-1', если значение отрицательное, '0', если значение равно нулю, '1', если значение положительное.
sin	Синус значения, где значение — это угол, выраженный в радианах.
sinh	Гиперболический синус значения, где значение — это угол, выраженный в радианах.
sqrt	Квадратный корень значения.
sum	Сумма значений указанных элементов данных.
tan	Тангенс значения.
truncate	Отсечь значение до указанного числа знаков после запятой.

Подробности функций

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми
- в качестве параметров допустимы выражения
- необязательные параметры функций (или части параметров) обозначаются угловыми скобками < >

abs(значение)

Абсолютное значение (по модулю) значения.

Параметр:

- **значение** — проверяемое значение

Например, абсолютное значение как «3», так и «-3» будет «3».

Пример:

```
abs(last(/  _  /  ))>10
```

acos(значение)

Арккосинус значения как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Значение должно быть в диапазоне от -1 до 1. Например, арккосинус значения «0.5» будет «2.0943951».

Пример:

```
acos(last(/  _  /  ))
```

asin(значение)

Арксинус значения как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Значение должно быть в диапазоне от -1 до 1. Например, арксинус значения «0.5» будет «-0.523598776».

Пример:

```
asin(last(/  _  /  ))
```

atan(значение)

Арктангенс значения как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Например, арктангенс значения «1» будет «0.785398163».

Пример:

```
atan(last(/  _  /  ))
```

atan2(значение,абсцисса)

Арктангенс указанных координат ординаты (значение) и абсцисса как угол, выраженный в радианах.

Параметры:

- **значение** — проверяемое значение;
- **абсцисса** — значение абсциссы.

Например, арктангенс координат, где ордината и абсцисса имеют значение «1», будет «2.21429744».

Пример:

```
atan2(last(/  _  /  ),2)
```

avg(<значение1>,<значение2>,...)

Среднее из значений, на которые ссылаются перечисленные элементы данных.

Параметр:

- **значениеX** — значение, возвращаемое другой функцией, работающей с историей элементов данных.

Пример:

```
avg(avg(/  _  /  ,1h),avg(/  _  2/  2,1h))
```

cbrt(значение)

Кубический корень из значения.

Параметр:

- **значение** — проверяемое значение

Например, кубический корень из «64» будет «4», а из «63» будет «3.97905721».

Пример:

```
cbrt(last(/  _  /  ))
```

ceil(значение)

Округлить значение вверх до ближайшего целого.

Параметр:

- **значение** — проверяемое значение

Например, «2.4» будет округлено до «3». Смотрите также floor().

Пример:

```
ceil(last(/  _  /  ))
```

cos(значение)

Косинус значения, где значение воспринимается как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Например, косинус значения «1» будет «0.54030230586».

Пример:

```
cos(last(/  _  /  ))
```

cosh(значение)

Гиперболический косинус значения. Возвращает значение как действительное число, не в научной нотации.

Параметр:

- **значение** — проверяемое значение

Например, гиперболический косинус значения «1» будет «1.54308063482».

Пример:

```
cosh(last(/  _  /  ))
```

cot(значение)

Котангенс значения, где значение воспринимается как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Например, котангенс значения «1» будет «0.54030230586».

Пример:

```
cot(last(/  _  /  ))
```

degrees(значение)

Конвертация значения из радиан в градусы.

Параметр:

- **значение** — проверяемое значение

Например, значение «1», сконвертированное в градусы, будет «57.2957795».

Пример:

```
degrees(last(/  _  /  ))
```

e

Число Эйлера (2.718281828459045).

Пример:

```
e()
```

exp(значение)

Число Эйлера (e) в степени значения.

Параметр:

- **значение** — проверяемое значение

Например, число Эйлера, возведённое в степень «2», будет «7.38905609893065».

Пример:

```
exp(last(/  _  /  ))
```

expm1(значение)

Число Эйлера (e) в степени значения, минус один.

Параметр:

- **значение** — проверяемое значение

Например, число Эйлера, возведённое в степень «2», после вычитания единицы будет «6.38905609893065».

Пример:

```
expm1(last(/  _  /  ))
```

floor(значение)

Округление вниз до ближайшего целого.

Параметр:

- **значение** — проверяемое значение

Например, «2.6» будет округлено вниз до «2». Смотрите также ceil().

Пример:

```
floor(last(/  _  /  ))
```

log(значение)

Натуральный логарифм.

Параметр:

- **значение** — проверяемое значение

Например, натуральный логарифм значения «2» будет «0.69314718055994529».

Пример:

```
log(last(/  _  /  ))
```

log10(значение)

Десятичный логарифм.

Параметр:

- **значение** — проверяемое значение

Например, десятичный логарифм значения «5» will be «0.69897000433».

Пример:

```
log10(last(/  _  /  ))
```

max(<значение1>,<значение2>,...)

Наибольшее из значений, на которые ссылаются перечисленные элементы данных.

Параметр:

- **значениеX** — значение, возвращаемое другой функцией, работающей с историей элементов данных.

Пример:

```
max(avg(/  _  /  ,1h),avg(/  _  2/  2,1h))
```

min(<значение1>,<значение2>,...)

Наименьшее из значений, на которые ссылаются перечисленные элементы данных.

Параметр:

- **значениеX** — значение, возвращаемое другой функцией, работающей с историей элементов данных.

Пример:

```
min(avg(/  _  /  ,1h),avg(/  _  2/  2,1h))
```

mod(значение,делитель)

Остаток от деления.

Параметры:

- **значение** — проверяемое значение;
- **делитель** — знаменатель деления.

Например, остаток от деления значения «5» с делителем «2» будет «1».

Пример:

```
mod(last(/  _  /  ),2)
```

pi

Константа пи (3.14159265358979).

Пример:

```
pi()
```

power(значение,степень)

Возведение в степень.

Параметры:

- **значение** — проверяемое значение;
- **степень** — используемый показатель степени.

Например, третья степень значения «2» будет «8».

Пример:

```
power(last(/  _  /  ),3)
```

radians(значение)

Конвертация значения из градусов в радианы.

Параметр:

- **значение** — проверяемое значение

Например, значение «1», сконвертированное в радианы, будет «0.0174532925».

Пример:

```
radians(last(/  _  /  ))
```

rand

Вернуть случайное целое число. Псевдослучайное число, сгенерированное с использованием времени в качестве seed (достаточно для математичеких целей, но не для криптографии).

Пример:

```
rand()
```

round(значение,дес_знаков)

Округлить значение до десятичных знаков.

Параметры:

- **значение** — проверяемое значение;
- **дес_знаков** — количество десятичных знаков для округления (0 также допустим).

Например, значение «2.5482», округлённое до 2 десятичных знаков, будет «2.55».

Пример:

```
round(last(/  _  /  ),2)
```

signum(значение)

Возвращает «-1» для отрицательного значения, «0» для нуля, «1» для положительного значения.

Параметр:

- **значение** — проверяемое значение.

Пример:

```
signum(last(/  _  /  ))
```

sin(значение)

Синус значения, воспринимаемого как угол, выраженный в радианах.

Параметр:

- **значение** — проверяемое значение

Например, синус значения «1» будет «0.8414709848».

Пример:

```
sin(last(/  _  /  ))
```

sinh(значение)

Гиперболический синус значения.

Параметр:

- **значение** — проверяемое значение

Например, гиперболический синус значения «1» будет «1.17520119364».

Пример:

```
sinh(last(/  _  /  ))
```

sqrt(значение)

Квадратный корень значения.
 Эта функция выдаст ошибку для отрицательного значения.

Параметр:

- **значение** — проверяемое значение

Например, квадратный корень из значения «3.5» will be «1.87082869339».

Пример:

```
sqrt(last(/  _  /  ))
```

sum(<значение1>,<значение2>,...)

Сумма значений, на которые ссылаются перечисленные элементы данных.

Параметр:

- **значениеX** — значение, возвращаемое другой функцией, работающей с историей элементов данных.

Пример:

```
sum(avg(/  _  /  ,1h),avg(/  _  2/  2,1h))
```

tan(значение)

Тангенс значения.

Параметр:

- **значение** — проверяемое значение

Например, тангенс значения «1» будет «1.55740772465».

Пример:

```
tan(last(/  _  /  ))
```

truncate(значение,дес_знаков)

Усечение значения до указанного числа десятичных знаков.

Параметры:

- **значение** — проверяемое значение;
- **дес_знаков** — количество десятичных знаков, оставляемых после усечения (0 также допустим).

Например, значение как «2.5482», усечённое до 2 десятичных знаков, будет «2.54».

Пример:

```
truncate(last(/  _  /  ),2)
```

См. все поддерживаемые функции.

Операторные функции

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы просмотреть подробные сведения.

Функция	Описание
between	Проверка принадлежности значения заданному диапазону.
in	Проверка равенства значения хотя бы одному из перечисленных значений.

Подробности функций

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми

- в качестве параметров допустимы выражения

between(значение,мин,макс)

Проверка того, попадает ли значение в заданный диапазон.
 Поддерживаемые типы значений: *Integer*, *Float*.
 Возвращает: 1 — в диапазоне; 0 — в противном случае.

Параметры:

- **значение** — проверяемое значение;
- **мин** — минимальное значение;
- **макс** — максимальное значение.

Пример:

```
between(last(/  _  /  ),1,10)=1 #          ,                  1    10
```

in(значение,значение1,значение2,...значениеN)

Проверка того, что значение равно хотя бы какому-нибудь из перечисленных значений.
 Поддерживаемые типы значений: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Возвращает: 1 — если равно; 0 — в противном случае.

Параметры:

- **значение** — проверяемое значение;
- **значение1,значение2,...значениеN** — перечисленные значения (строковые значения должны быть в двойных кавычках).

Значение сравнивается с перечисленными значениями как число, если все эти значения могут быть преобразованы в число; в противном случае сравнивается как строка.

Примеры:

```
in(last(/  _  /  ),5,10)=1 #          ,                      5,    10
in("text",last(/  _  /  ),last(/  _  /  ,#2))=1 #          ,                          «text»
```

См. все поддерживаемые функции.

Прогностические функции

Все функции, перечисленные здесь, поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Функции перечислены без дополнительной информации. Щелкните функцию, чтобы просмотреть полные сведения.

Function	Description
forecast	Будущее значение, max, min, delta или avg элемента данных.
timeleft	Время в секундах, необходимое элементу данных для достижения указанного порога.

Общие параметры

- `/host/key` - общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети.
- `(sec|#num)<:time shift>` - общий второй параметр для функций, ссылающихся на историю элемента данных узла сети, где:
  - **sec** - максимальный период вычисления в секундах (можно использовать суффиксы времени) или
  - **#num** - максимальный диапазон вычисления в последних собранных значениях (если перед ним стоит символ решетки).
  - **time shift** (необязательно) - позволяет сместить точку вычисления назад во времени. Подробнее о задании временного сдвига см. здесь.

Подробности функций

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются < >
- Параметры, специфичные для функций, описаны с каждой функцией

- Параметры `/host/key` и `(sec|#num)<:time shift>` никогда не должны заключаться в кавычки

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

Будущее значение, максимальное, минимальное значение, дельта или среднее значение элемента данных.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры;
- **time** - горизонт прогнозирования в секундах (например, 1, 60) (можно использовать суффиксы времени (например, `30s`, `1m`)); поддерживаются отрицательные значения;
- **fit** (необязательный параметр; должен быть заключён в двойные кавычки) - функция, используемая для аппроксимации исторических данных. Поддерживаются следующие функции:
linear - линейная функция (по умолчанию)
polynomialN - полином степени N (1 <= N <= 6)
exponential - экспоненциальная функция
logarithmic - логарифмическая функция
power - степенная функция
Обратите внимание, что *polynomial1* эквивалентен *linear*;
- **mode** (необязательный параметр; должен быть заключён в двойные кавычки) - требуемый результат. Поддерживаются следующие режимы:
value - значение (по умолчанию)
max - максимальное значение
min - минимальное значение
delta - *max-min*
avg - среднее значение
Обратите внимание, что *value* оценивает значение элемента данных в момент `now + time`; режимы *max*, *min*, *delta* и *avg* оценивают значение элемента данных на интервале между `now` и `now + time`.

Комментарии:

- Если возвращаемое значение больше 1.7976931348623158E+308 или меньше -1.7976931348623158E+308, оно соответственно ограничивается значением 1.7976931348623158E+308 или -1.7976931348623158E+308;
- Становится неподдерживаемым только при неправильном использовании в выражении (неверный тип элемента данных, недопустимые параметры); в противном случае при возникновении ошибок возвращает -1;
- Дополнительную информацию см. в разделе функции прогнозирования для триггеров.

Примеры:

```
forecast(/host/key,#10,1h) #                                    10
forecast(/host/key,1h,30m) #                          30
forecast(/host/key,1h:now-1d,12h) #                         12                  ,
forecast(/host/key,1h,10m,"exponential") #                    10
forecast(/host/key,1h,2h,"polynomial3","max") #                ,
forecast(/host/key,#2,-20m) #                  20                          (           ,          1
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Время в секундах, необходимое элементу данных для достижения указанного порогового значения.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры;
- **threshold** - значение, которого необходимо достичь (можно использовать суффиксы единиц измерения);
- **fit** (необязательный параметр; должен быть заключен в двойные кавычки) - см. forecast().

Комментарии:

- Если возвращаемое значение превышает 1.7976931348623158E+308, оно ограничивается значением 1.7976931348623158E+30
- Возвращает 1.7976931348623158E+308, если пороговое значение недостижимо;
- Становится неподдерживаемым только при неправильном использовании в выражении (неверный тип элемента данных, недопустимые параметры); в противном случае при ошибках возвращает -1;
- Дополнительную информацию см. в разделе о прогнозирующих функциях триггеров.

Примеры:

```
timeleft(/host/key,#10,0) #                                     10
timeleft(/host/key,1h,100) #                            100
timeleft(/host/key,1h:now-1d,100) #                           100                 ,
timeleft(/host/key,1h,200,"polynomial2") #                    200                           ,
```

См. все поддерживаемые функции.

Прогнозирующие функции триггеров

Обзор

Иногда появляются признаки надвигающейся проблемы. Эти признаки можно обнаружить, чтобы заранее принять меры для предотвращения проблемы или хотя бы минимизации ее последствий.

В Zabbix есть инструменты для прогнозирования будущего поведения контролируемой системы на основе исторических данных. Эти инструменты реализованы с помощью прогностических функций триггеров.

Функции

Перед настройкой триггера необходимо определить, что считается проблемным состоянием и сколько времени требуется для принятия мер. Затем есть два способа настроить триггер, сигнализирующий о потенциальной нежелательной ситуации. Первый: триггер должен сработать, когда ожидается, что система будет в проблемном состоянии через «время на реагирование». Второй: триггер должен сработать, когда система достигнет проблемного состояния менее чем через «время на реагирование». Соответствующие функции триггера для использования — **forecast** и **timeleft**. Обратите внимание, что лежащий в основе статистический анализ по сути идентичен для обеих функций. Вы можете настроить триггер любым удобным для вас способом и получить схожие результаты.

Параметры

Обе функции используют почти одинаковый набор параметров. Для справки используйте список поддерживаемых функций.

Интервал времени

Прежде всего, следует указать исторический период, который Zabbix должен проанализировать, чтобы построить прогноз. Это делается привычным способом с помощью параметра `time period` и необязательного сдвига по времени, так же, как и в функциях **avg**, **count**, **delta**, **max**, **min** и **sum**.

Горизонт прогнозирования

(только для **forecast**)
 Параметр `time` указывает, насколько далеко в будущее Zabbix должен экстраполировать зависимости, обнаруженные в исторических данных. Независимо от того, используете вы `time_shift` или нет, `time` всегда отсчитывается от текущего момента.

Порог для достижения

(только для **timeleft**)
 Параметр `threshold` задаёт значение, которого анализируемый элемент данных должен достичь, независимо от того, сверху или снизу. После того как мы определили f(t) (см. ниже), следует решить уравнение f(t) = `threshold` и вернуть корень, который ближе к текущему моменту и находится справа от него, либо 1.7976931348623158E+308, если такого корня нет.

Note:
Когда значения элемента данных приближаются к порогу, а затем пересекают его, **timeleft** предполагает, что пересечение уже произошло в прошлом, и поэтому переключается на следующее пересечение с уровнем `threshold`, если таковое имеется. Рекомендуется использовать прогнозы как дополнение к обычной диагностике проблем, а не как её замену.[1]

Функции аппроксимации

По умолчанию `fit` — это *линейная* функция. Но если ваша отслеживаемая система более сложная, у вас есть больше вариантов на выбор.

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$
exponential	$x = a*exp(b*t)$
logarithmic	$x = a + b*log(t)$
power	$x = a*t^b$

Режимы

(только для **forecast**)
 Каждый раз, когда вычисляется функция триггера, она получает данные из указанного периода истории и подбирает указанную функцию к данным. Таким образом, если данные немного отличаются, подобранная функция также будет немного отличаться. Если просто вычислить значение подобранной функции в указанный момент времени в будущем, вы ничего не узнаете о том, как ожидается поведение анализируемого элемента данных в промежутке между текущим моментом и этим моментом в будущем. Для некоторых параметров `fit` (например, *polynomial*) простое значение из будущего может вводить в заблуждение.

mode	Результат **forecast**
value	f(now + time)
max	$\max_{now <= t <= now + time}$ f(t)
min	$\min_{now <= t <= now + time}$ f(t)
delta	*max - min*
avg	среднее значение f(t) (now <= t <= now + time) согласно определению

Подробности

Чтобы избежать вычислений с огромными числами, мы рассматриваем временную метку первого значения в указанном периоде плюс 1 нс как новое нулевое время (текущее время эпохи имеет порядок 10^9, квадрат эпохи — 10^{18}, двойная точность составляет около 10^{-16}). 1 нс добавляется, чтобы обеспечить все положительные значения времени для *логарифмической* и *степенной* аппроксимации, которые включают вычисление log(t). Сдвиг времени не влияет на *линейную*, *полиномиальную*, *экспоненциальную* аппроксимацию (кроме более простых и точных вычислений), но изменяет форму *логарифмической* и *степенной* функций.

Возможные ошибки

Функции возвращают -1 в следующих случаях:

- указанный период оценки не содержит данных;
- результат математической операции не определён[3];
- численные осложнения (к сожалению, для некоторых наборов входных данных диапазон и точность формата чисел с плавающей точкой двойной точности оказываются недостаточными)[4].

Note:
Предупреждения или ошибки не выдаются, если выбранная аппроксимация плохо описывает предоставленные данные или если данных просто слишком мало для точного прогнозирования.

Примеры и обработка ошибок

Чтобы получить предупреждение, когда на вашем узле сети вот-вот закончится свободное место на диске, вы можете использовать такое выражение триггера:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

Однако в дело может вмешаться код ошибки -1 и перевести ваш триггер в проблемное состояние. В целом это хорошо, потому что вы получаете предупреждение о том, что ваши прогнозы работают некорректно, и вам следует изучить их более тщательно, чтобы выяснить причину. Но иногда это плохо, потому что -1 может просто означать, что за последний час не было получено данных о свободном месте на диске узла сети. Если вы получаете слишком много ложноположительных оповещений, рассмотрите возможность использования более сложного выражения триггера [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

С **forecast** ситуация немного сложнее. Прежде всего, -1 может как перевести триггер в проблемное состояние, так и не сделать этого, в зависимости от того, есть ли у вас выражение вида `forecast(/host/item,(...))<...` или вида `forecast(/host/item,(...))>...`

Кроме того, -1 может быть допустимым прогнозом, если для элемента данных нормально иметь отрицательное значение. Но вероятность такой ситуации в реальных условиях пренебрежимо мала (см. как работает оператор =). Поэтому добавьте `... or forecast(/host/item,(...))=-1` или `... and forecast(/host/item,(...))<>-1`, если вы соответственно хотите или не хотите рассматривать -1 как проблему.

Сноски

[1] Например, простой триггер вида `timeleft(/host/item,1h,X) < 1h` может перейти в проблемное состояние, когда значение элемента данных приближается к X, а затем внезапно восстановиться, как только значение X будет достигнуто. Если проблема заключается в том, что значение элемента данных ниже X, используйте: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Если проблема заключается в том, что значение элемента данных выше X, используйте: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] Степень полинома может быть от 1 до 6, *polynomial1* эквивалентен *linear*. Однако используйте полиномы более высокой степени с осторожностью. Если период оценки содержит меньше точек, чем требуется для определения коэффициентов полинома, степень полинома будет понижена (например, запрошен *polynomial5*, но есть только 4 точки, поэтому будет аппроксимирован *polynomial3*).

[3] Например, аппроксимация функциями *exponential* или *power* включает вычисление log() от значений элемента данных. Если данные содержат нули или отрицательные числа, вы получите ошибку, поскольку log() определён только для положительных значений.

[4] Для аппроксимаций *linear*, *exponential*, *logarithmic* и *power* все необходимые вычисления можно записать явно. Для *polynomial* без дополнительных шагов можно вычислить только *value*. Вычисление *avg* включает нахождение первообразной полинома (аналитически). Вычисление *max*, *min* и *delta* включает нахождение производной полинома (аналитически) и поиск её корней (численно). Решение f(t) = 0 включает поиск корней полинома (численно).

[5] Но в этом случае -1 может привести к восстановлению вашего триггера из проблемного состояния. Чтобы полностью защититься, используйте: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)`

Строковые функции

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Формулах вычисляемых элементов данных
- Макросах выражений

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы просмотреть подробные сведения.

Функция	Описание
ascii	Код ASCII крайнего левого символа значения.
bitlength	Длина значения в битах.
bytelength	Длина значения в байтах.
char	Возвращает символ, интерпретируя значение как код ASCII.
concat	Строка, полученная путем объединения значений указанных элементов данных или постоянных значений.
contains	Возвращает 1, если значение содержит подстроку, указанную в параметре `search value`, и 0 в противном случае.
insert	Вставляет указанные символы или пробелы в строку, начиная с указанной позиции.
jsonpath	Возвращает результат JSONPath.
left	Возвращает крайние левые символы значения.
length	Длина значения в символах.
ltrim	Удаляет указанные символы из начала строки.
mid	Возвращает подстроку из N символов, начиная с позиции символа, указанной в параметре 'start'. Эта функция является псевдонимом функции `substring()`.
repeat	Повторяет строку.
replace	Находит шаблон в значении и заменяет его указанной строкой.
right	Возвращает крайние правые символы значения.
rtrim	Удаляет указанные символы из конца строки.
substring	Возвращает подстроку из N символов, начиная с позиции символа, указанной в параметре 'start'.
trim	Удаляет указанные символы из начала и конца строки.
xmlxpath	Возвращает результат XML XPath.

Подробности функций

Некоторые общие замечания о параметрах функций:

- параметры функций разделяются запятыми
- в качестве параметров допустимы выражения
- строковые параметры должны заключаться в двойные кавычки; в противном случае они могут быть интерпретированы неверно
- необязательные параметры функций (или части параметров) обозначаются угловыми скобками< >

ascii(значение)

Код ASCII самого левого символа значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметр:

- **значение** — проверяемое значение

Например, значение «Abc» вернёт «65» (код ASCII для буквы «A»).

Пример:

```
ascii(last(/  _  /  ))
```

bitlength(значение)

Длина значения в битах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Integer*.

Параметр:

- **значение** — проверяемое значение

Пример:

```
bitlength(last(/  _  /  ))
```

bytelength(значение)

Длина значения в байтах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Integer*.

Параметр:

- **значение** — проверяемое значение

Пример:

```
bytelength(last(/  _  /  ))
```

char(значение)

Вернуть символ, интерпретируя значение как код ASCII.
 Поддерживаемые типы значений: *Integer*.

Параметр:

- **значение** — проверяемое значение

Значение должно быть в диапазоне 0-255. Например, значение «65» (интерпретируемое как код ASCII) вернёт «A».

Пример:

```
char(last(/  _  /  ))
```

concat(<значение1>,<значение2>,...)

Строка, получаемая в результате конкатенации упоминаемых значений элементов данных либо констант.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Float*, *Integer*.

Параметр:

- **значениеX** — значение, возвращаемое одной из функций истории, либо константа (string, integer или float number). Должно содержать не менее двух параметров.

Например, значение вроде «Zab», объединённое с «bix» (строки-константы), вернёт «Zabbix».

Примеры:

```
concat(last(/  _  /  ),"bix")
concat("1 min: ",last(/  _  /system.cpu.load[all,avg1]),", 15 min: ",last(/  _  /system.cpu.load[all,avg1
```

contains(value, search value)

Возвращает 1, если значение содержит подстроку, указанную в `search value`, и 0 в противном случае.| Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** - проверяемое значение;
- **search value** - искомое значение.

Примечания:

- Поиск чувствителен к регистру.
- Если значение корректно, а `search value` пусто, функция возвращает 1.

Например, функция возвращает 1, если значение содержит «Zabbix».

Пример:

```
contains(last(/host/key),"Zabbix")
```

insert(значение,старт,длина,замена)

Вставить указанные символы или пробелы в символьную строку, начиная с указанной позиции в строке.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **старт** — начальная позиция;
- **длина** — заменяемых позиций;
- **замена** — подставляемая строка.

Например, значение вроде «Zabbbix» будет заменено на «Zabbix», если «bb» (начиная с позиции 3, заменяемых позиций 2) заменяется строкой «b».

Пример:

```
insert(last(/  _  /  ),3,2,"b")
```

jsonpath(значение,путь,<умолчание>)

Вернуть результат JSONPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **путь** — путь (должен быть заключён в кавычки);
- **умолчание** — необязательное резервное значение на случай, если запрос JSONPath не возвращает данных. Обратите внимание, что при других ошибках возвращается сбой (например, «unsupported construct»).

Пример:

```
jsonpath(last(/  _  /proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(значение,кол-во)

Вернуть самые левые символы значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **кол-во** — количество возвращаемых символов.

Например, можно вернуть «Zab» из «Zabbix», указывая для возврата 3 крайних слева символа. См. также right().

Пример:

```
left(last(/  _  /  ),3) #
```

length(value)

Длина значения в символах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметр:

- **значение** — проверяемое значение.

Примеры:

```
length(last(/  _  /  )) #
length(last(/  _  /  ,#3)) #
length(last(/  _  /  ,#1:now-1d)) #
```

ltrim(значение,<символы>)

Удалить указанные символы с начала строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **символы** (опционально) — укажите удаляемые символы.

По умолчанию удаляются пробелы (если не указано опциональных символов). См. также: rtrim(), trim().

Примеры:

```
ltrim(last(/  _  /  )) #
ltrim(last(/  _  /  ),"Z") #             «Z»
ltrim(last(/  _  /  )," Z") #                «Z»
```

mid(value,start,length)

Возвращает подстроку из N символов, начиная с позиции символа, указанной в параметре «start».
Эта функция является псевдонимом функции `substring()`.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** - проверяемое значение;
- **start** - начальная позиция подстроки;
- **length** - количество позиций, возвращаемых в подстроке.

Например, из значения «Zabbix» можно вернуть «abbi», если начальная позиция равна 2, а количество возвращаемых позиций - 4.

Пример:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(значение,кол-во)

Повторить строку.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **кол-во** — количество повторений.

Пример:

```
repeat(last(/  _  /  ),2) #
```

replace(значение,образец,замена)

Найти в значении образец и заменить его заменой. Будут заменены все вхождения образца.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **образец** — искомый образец;
- **замена** — строка, которой заменяется образец.

Пример:

```
replace(last(/  _  /  ),"ibb","abb") —          «ibb»   «abb»
```

right(value,count)

Вернуть самые правые символы значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **кол-во** — количество возвращаемых символов.

Например, можно вернуть «bix» из «Zabbix», если указать возвращать 3 символа справа. См. также left().

Пример:

```
right(last(/  _  /  ),3) #
```

rtrim(значение,<символы>)

Удалить указанные символы с конца строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **символы** (опционально) — укажите удаляемые символы.

По умолчанию удаляются пробелы (если не указано опциональных символов). См. также: ltrim(), trim().

Примеры:

```
rtrim(last(/  _  /  )) #
rtrim(last(/  _  /  ),"x") #             «x»
rtrim(last(/  _  /  ),"x ") #             «x»
```

substring(value,start,length)

Возвращает подстроку из N символов, начиная с позиции символа, указанной в параметре «start».
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** - проверяемое значение;
- **start** - начальная позиция подстроки;
- **length** - количество позиций, возвращаемых в подстроке.

Например, из значения «Zabbix» можно вернуть «abbi», если начальная позиция равна 2, а количество возвращаемых позиций - 4.

Пример:

```
substring(last(/host/key),2,4)="abbi"
```

trim(значение,<символы>)

Удалить указанные символы с начала и конца строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **символы** (опционально) — укажите удаляемые символы.

По умолчанию удаляются пробелы (если не указано опциональных символов). См. также: ltrim(), rtrim().

Примеры:

```
trim(last(/  _  /  )) #
trim(last(/  _  /  ),"_") #     «_»
```

xmlxpath(значение,путь,<умолчание>)

Вернуть результат XML XPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **значение** — проверяемое значение;
- **путь** — путь (должен быть заключён в кавычки);
- **умолчание** — необязательное резервное значение на случай, если запрос XML XPath возвращает пустой набор узлов. Оно не будет возвращено, если пустой результат не является набором узлов (т.е. пустой строкой). При других ошибках возвращается сбой (например, «invalid expression»).

Пример:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

См. все поддерживаемые функции.

Лучшие практики для триггеров

Триггеры — это мощный инструмент, но они также могут создавать нежелательный шум оповещений. Чтобы видеть больше реальных сигналов и меньше шума, следуйте этим рекомендациям:

1. Снизьте чувствительность триггеров. Вместо оповещения по последнему значению (слишком высокому/слишком низкому) анализируйте среднее значение за длительный период, используя функции **avg**, **min** и **max**.
2. Рассмотрите возможность использования функции **percentile** (установите ее на 95% или 5%) в триггерах, если хотите избежать оповещений о случайных всплесках и падениях.
3. Используйте гистерезис, чтобы избежать дребезга триггера — частых изменений состояния триггера (OK ↔ Problem). Континуум для состояния проблемы можно определить, добавив выражение восстановления (отдельное условие для устранения проблемы).
4. Используйте зависимости триггеров, чтобы избежать оповещений, не связанных с первопричиной.
5. Используйте серьезность триггера, чтобы оповещать только о более серьезных проблемах.
6. Определяйте окна обслуживания.

Гистерезис

Иногда между состояниями проблемы и восстановления требуется интервал, а не просто пороговое значение. Например, если мы хотим определить триггер, который сообщает о проблеме, когда температура в серверной превышает 20°C, и хотим, чтобы он оставался в состоянии проблемы до тех пор, пока температура не опустится ниже 15°C, простого порога триггера на уровне 20°C будет недостаточно.

Вместо этого сначала нужно определить выражение триггера для события проблемы (температура выше 20°C). Затем нужно определить дополнительное условие восстановления (температура ниже 15°C). Это делается путем определения *выражения восстановления* при настройке триггера.

В этом случае восстановление после проблемы происходит в два этапа:

- Сначала выражение проблемы (температура выше 20°C) должно принять значение FALSE
- Затем выражение восстановления (температура ниже 15°C) должно принять значение TRUE

Выражение восстановления вычисляется только после того, как событие проблемы устранено. Само по себе значение TRUE для выражения восстановления не устраняет проблему, если выражение проблемы по-прежнему имеет значение TRUE!

Пример

Температура в серверной слишком высокая.

Выражение проблемы:

```
last(/server/temp)>20
```

Выражение восстановления:

```
last(/server/temp)<=15
```

> **Note:**
> Использовать макрос {TRIGGER.VALUE} в выражении восстановления нецелесообразно, поскольку это выражение вычисляется только тогда, когда триггер находится в состоянии ”Проблема”. Следовательно, при вычислении выражения {TRIGGER.VALUE} всегда будет принимать значение ”1” (что указывает на состояние ”Проблема”).

Зависимости триггеров

Зависимости триггеров можно использовать, чтобы избежать оповещений, которые не связаны с первопричиной.

См. все рекомендации.

Обзор

Иногда доступность одного узла сети зависит от другого. Сервер, находящийся за маршрутизатором, станет недоступен, если маршрутизатор выйдет из строя. Если для обоих настроены триггеры, вы можете получить уведомления о недоступности двух узлов сети, хотя проблема была только в маршрутизаторе.

В таких случаях может быть полезна зависимость между узлами сети. При настроенной зависимости уведомления о зависимых объектах могут подавляться, а отправляться будет только уведомление о корневой проблеме.

Хотя Zabbix не поддерживает зависимости между узлами сети напрямую, их можно определить с помощью другого, более гибкого метода — зависимостей триггеров. Триггер может зависеть от одного или нескольких других триггеров.

В нашем простом примере мы открываем форму настройки триггера сервера и указываем, что он зависит от соответствующего триггера маршрутизатора. При такой зависимости триггер сервера не изменит свое состояние, пока триггер, от которого он зависит, находится в состоянии «PROBLEM». Поэтому зависимые действия выполняться не будут и уведомления отправляться не будут.

Если сервер и маршрутизатор недоступны, а между триггером «сервер недоступен» и триггером «маршрутизатор недоступен» настроена зависимость, Zabbix не будет выполнять действия для зависимого триггера.

Пока родительский триггер находится в состоянии PROBLEM, зависимые от него триггеры могут сообщать значения, которым нельзя доверять. Поэтому зависимые триггеры не будут повторно вычисляться, пока родительский триггер (маршрутизатор в приведенном выше примере):

- не перейдет из состояния «PROBLEM» в состояние «OK»;
- не изменит свое состояние с «PROBLEM» на «UNKNOWN»;
- не будет закрыт вручную, с помощью корреляции или с использованием функций date and time и/или nodata();
- не будет разрешен значением элемента данных, не участвующего в зависимом триггере;

- не будет отключен, либо у него не будет отключенного элемента данных или отключенного узла сети

Во всех перечисленных выше случаях зависимый триггер (сервер) будет повторно вычислен только после получения для него нового показателя. Это означает, что зависимый триггер может обновиться не сразу.

Также:

- Зависимость триггера может быть добавлена от триггера любого узла сети к любому другому триггеру узла сети, если это не приводит к циклической зависимости.
- Зависимость триггера может быть добавлена от одного шаблона к другому. Если какой-либо триггер из шаблона A зависит от какого-либо триггера из шаблона B, шаблон A можно связать с узлом сети (или другим шаблоном) только вместе с шаблоном B, тогда как шаблон B можно связать с узлом сети (или другим шаблоном) отдельно.
- Зависимость триггера может быть добавлена от триггера шаблона к триггеру узла сети. В этом случае связывание такого шаблона с узлом сети создаст триггер узла сети, зависящий от того же триггера шаблона, от которого зависел исходный триггер. Это позволяет, например, создать шаблон, в котором некоторые триггеры зависят от триггеров маршрутизатора (узла сети). Все узлы сети, связанные с этим шаблоном, будут зависеть от указанного маршрутизатора.
- Зависимость триггера не может быть добавлена от триггера узла сети к триггеру шаблона.
- Зависимость триггера может быть добавлена от прототипа триггера к другому прототипу триггера (в рамках того же правила низкоуровневого обнаружения) или к реальному триггеру. Прототип триггера не может зависеть от прототипа триггера из другого правила LLD или от триггера, созданного на основе прототипа триггера. Прототип триггера узла сети не может зависеть от триггера из шаблона.

Настройка

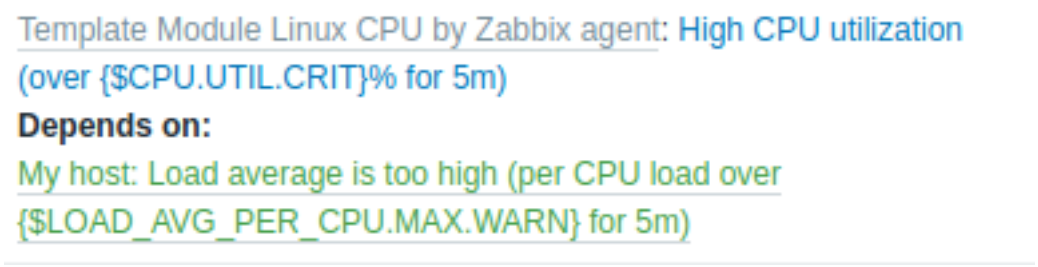
Чтобы определить зависимость:

1. Откройте вкладку *Зависимости* в форме настройки триггера.
2. Нажмите *Добавить* в разделе *Зависимости* и выберите один или несколько триггеров, от которых будет зависеть данный триггер.



3. Нажмите *Обновить*.

Теперь в списке у триггера отображается информация о его зависимости.



Пример нескольких зависимостей

Например, узел сети Host находится за Router2, а Router2 находится за Router1.

```
Zabbix - Router1 - Router2 - Host
```

Если Router1 недоступен, то, очевидно, Host и Router2 тоже недоступны, однако получение трех уведомлений о том, что Host, Router1 и Router2 недоступны, является избыточным.

Поэтому в этом случае мы определяем две зависимости:

```
    'Host is down'              'Router2 is down'
    'Router2 is down'              'Router1 is down'
```

Перед изменением состояния триггера 'Host is down' Zabbix проверит соответствующие зависимости триггеров. Если такие зависимости найдены и один из этих триггеров находится в состоянии **Problem**, то состояние триггера не будет изменено, действия не будут выполнены и уведомления не будут отправлены.

Zabbix выполняет эту проверку рекурсивно. Если Router1 или Router2 недоступен, триггер Host не будет обновлен.

Серьезность триггера

Серьезность триггера можно использовать, чтобы избежать лишних оповещений о менее серьезных проблемах.

См. все best practices.

Обзор

Серьезность отражает относительную важность триггера.



Назначение серьезности полезно не только для визуального отображения, но и для предотвращения уведомлений о незначительных проблемах.

Серьезности триггеров используются для:

- Визуального отображения (разный цвет для каждой созданной проблемы на основе исходной серьезности триггера)
- Ограничения уведомлений на основе серьезности (см. action conditions)
- Пользовательских медиа - разные медиа (канал уведомлений) для разных серьезностей (например, SMS для триггеров с серьезностью *High* и *Disaster*, и Email для триггеров с другими серьезностями)
- Звука в глобальных оповещениях - разные звуки для разных серьезностей

По умолчанию поддерживаются следующие серьезности триггеров:

Severity	Color	Description
Not classified	Gray	Можно использовать, когда уровень серьезности события неизвестен, не определен, не входит в обычную область мониторинга и т. д., например, при начальной настройке, как временное значение для будущей оценки или в рамках процесса интеграции.
Information	Light blue	Можно использовать для информационных событий, которые не требуют немедленного внимания, но при этом могут предоставлять полезные сведения.
Warning	Yellow	Можно использовать для обозначения потенциальной проблемы, которая может потребовать проверки или действий, но не является критической.
Average	Orange	Можно использовать для обозначения значительной проблемы, которую следует устранить в относительно короткие сроки, чтобы предотвратить дальнейшие неполадки.
High	Light red	Можно использовать для обозначения критических проблем, требующих немедленного внимания, чтобы избежать серьезных сбоев.
Disaster	Red	Можно использовать для обозначения серьезного инцидента, требующего немедленных действий, чтобы предотвратить, например, сбои системы или потерю данных.

Названия и цвета серьезностей триггеров можно **настраивать** в *Administration* > *General* > *Trigger displaying options*. Цвета общие для всех тем веб-интерфейса.

Перевод пользовательских имен важности

> **Attention:**
> Если используются переводы веб-интерфейса Zabbix, пользовательские имена важности по умолчанию будут переопределять переведенные имена.

Имена важности триггеров по умолчанию доступны для перевода во всех локалях.
Если имя важности изменено, пользовательское имя используется во всех локалях, и требуется дополнительный ручной перевод.

Процедура перевода пользовательского имени важности:

- задайте требуемое пользовательское имя важности, например, 'Important'
- отредактируйте <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- добавьте 2 строки:

```
msgid "Important"
msgstr "<         >"
```

и сохраните файл.

- создайте файлы .mo, как описано в <frontend_dir>/locale/README

Здесь **msgid** должен соответствовать новому пользовательскому имени важности, а **msgstr** должен содержать его перевод для конкретного языка.

Эту процедуру необходимо выполнять после каждого изменения имени важности.

Визуализация

Обзор

Визуализация предоставляет различные способы отображения данных мониторинга, собранных Zabbix. Графики показывают динамику изменений и метрики, карты сети отображают топологию инфраструктуры, а панели объединяют настраиваемые виджеты для предоставления всестороннего представления данных мониторинга.

Инструменты визуализации

- Графики — просматривайте значения элементов данных на простых автоматически сгенерированных графиках, создавайте собственные графики или сравнивайте несколько элементов данных в ситуационных графиках.
- Карты сети — создавайте визуальные представления сетевой инфраструктуры с настраиваемыми элементами и индикаторами.
- Панели — настраивайте макеты, комбинируя различные виджеты панели, которые визуализируют данные в разных форматах.

Графики

Обзор

При большом объеме данных, поступающих в Zabbix, пользователям гораздо удобнее видеть визуальное представление происходящего, а не только числовые значения.

Именно для этого используются графики. Графики позволяют с первого взгляда оценить поток данных, сопоставить проблемы, определить, когда что-то началось, или показать, когда ситуация может перерасти в проблему.

Zabbix предоставляет пользователям:

- Встроенные простые графики данных одного элемента данных.
- Возможность создавать более сложные пользовательские графики.
- Быстрый доступ к сравнению нескольких элементов данных в графиках по запросу.
- Современные настраиваемые векторные графики и круговые диаграммы.

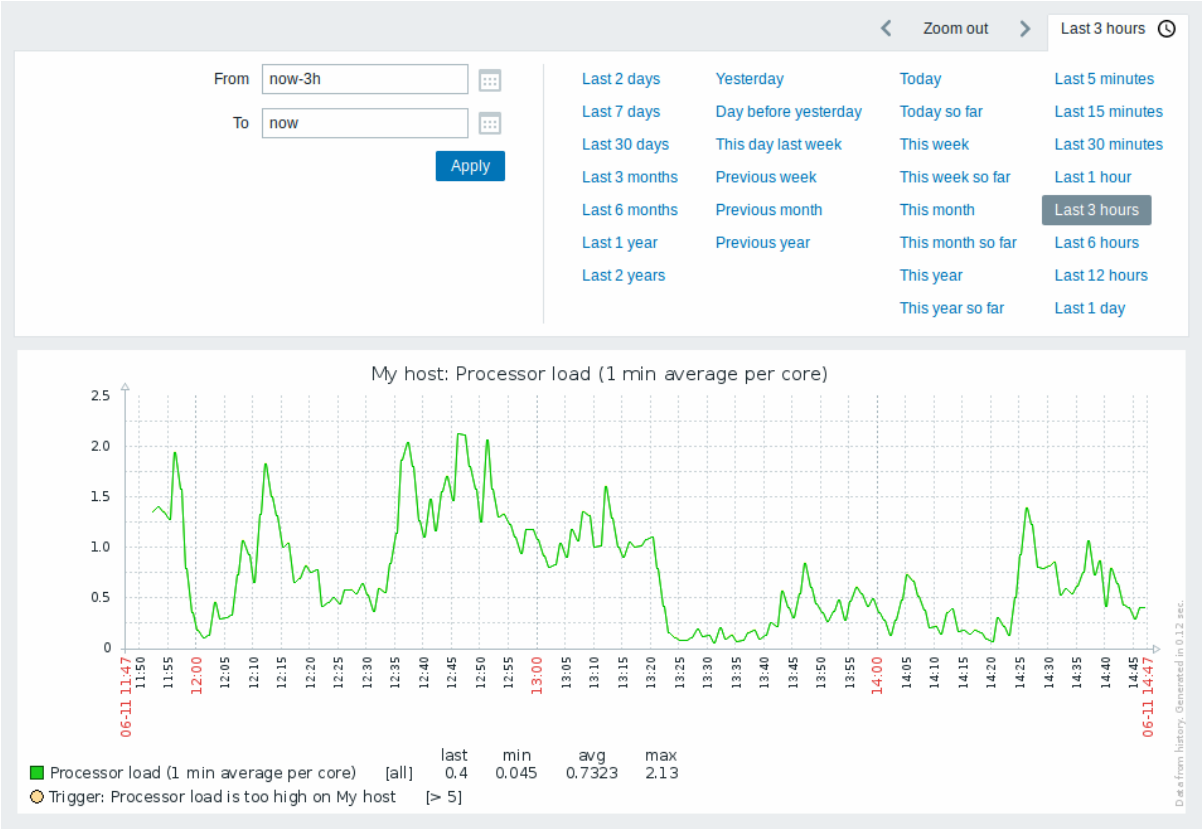
Простые графики

Обзор

Простые графики предназначены для визуализации данных, собранных элементами данных.

Чтобы просматривать простые графики, пользователю не требуется выполнять настройку. Они доступны в Zabbix без дополнительных действий.

Перейдите в раздел *Мониторинг* > *Последние данные* и нажмите ссылку *График* для соответствующего элемента данных — после этого отобразится график.



Note:
Простые графики доступны для всех числовых элементов данных. Для текстовых элементов данных в разделе *Мониторинг* > *Последние данные* доступна ссылка *История*.

Выбор периода времени

Расположенный над графиком выбор периода времени позволяет одним щелчком мыши выбрать часто используемые периоды.

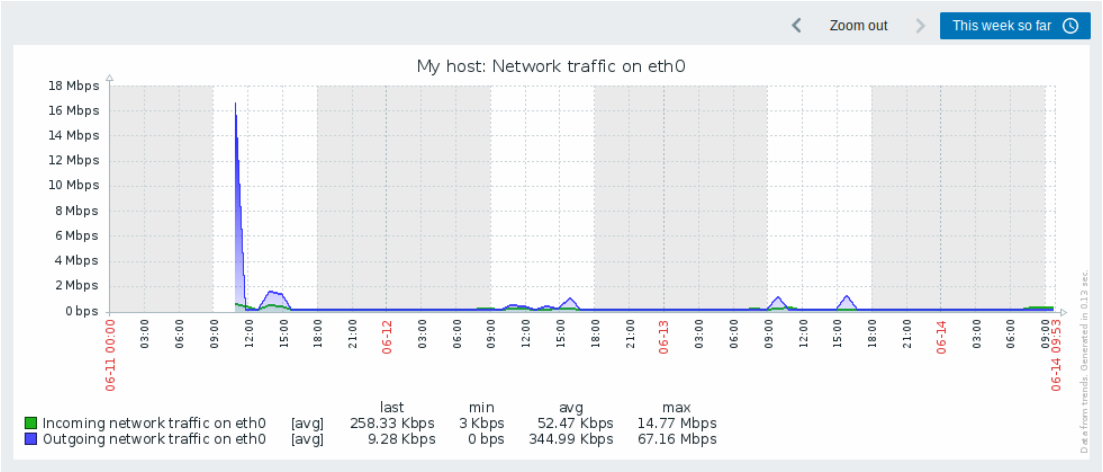
Дополнительную информацию см. в разделе Выбор периода времени и узла сети.

Недавние данные vs длительные периоды

Для самых последних данных рисуется **одиночная** линия, соединяющая каждое полученное значение. Одиночная линия рисуется до тех пор, пока доступен хотя бы один горизонтальный пиксель для одного значения.

Для данных, показывающих длительный период, рисуется **три линии** — тёмно-зелёная линия отображает среднее значение, тогда как светло-розовая и светло-зелёная линии отображают максимальное и минимальное значения в данный момент времени. Пространство между максимумами и минимумами заполняется жёлтым цветом.

Рабочее время (рабочие дни) отображается на графиках белым фоном, а нерабочее время отображается серым (в теме *Оригинальная синяя*, по умолчанию в веб-интерфейсе).

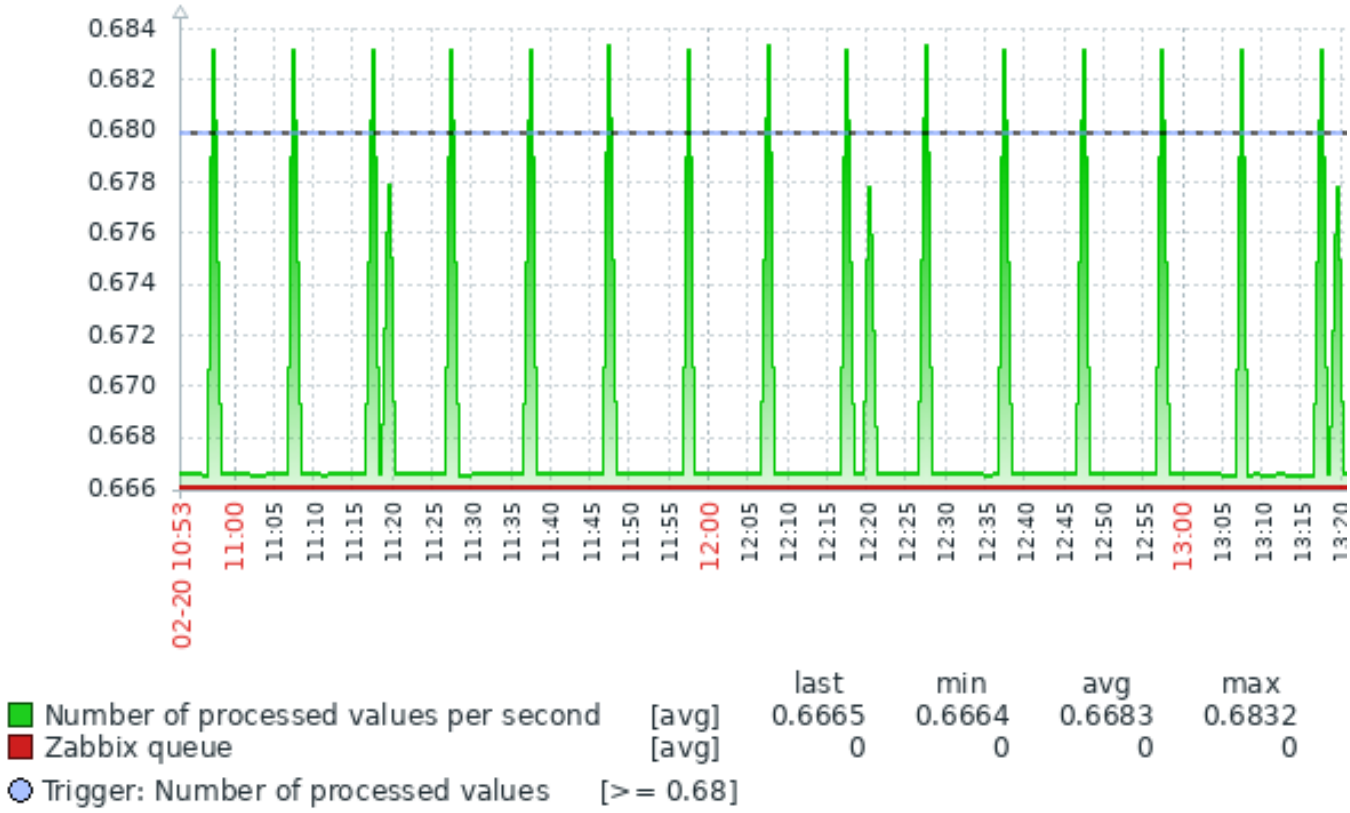


Рабочее время всегда отображается на простых графиках, тогда как отображение рабочего времени на пользовательских графиках является предпочтением пользователя.

Рабочее время не отображается, если график показывает период более 3 месяцев.

Линии триггеров

Простые триггеры отображаются линиями с чёрным пунктиром поверх цвета важности триггера — обратите внимание на синюю линию на графике и на информацию о триггере, отображаемую в легенде. На графике может отображаться до 3 линий триггеров; если триггеров больше, тогда приоритет отдается триггерам с меньшей важностью. В простых графиках триггеры отображаются всегда, в то время как отображение триггеров в пользовательских графиках является предпочтением пользователя.



Построение на основе истории/трендов

Графики могут строиться на основе истории или трендов элемента данных.

Для пользователей, у которых активирован режим отладки веб-интерфейса, в правом нижнем углу графика отображается серая вертикальная подпись, указывающая источник данных.

На выбор истории или трендов влияют несколько факторов:

- Срок хранения истории элемента данных. Например, история элемента данных может храниться в течение 14 дней. В этом случае данные старше четырнадцати дней будут поступать из трендов.

- Плотность данных на графике. Если количество секунд, отображаемых в одном горизонтальном пикселе графика, превышает 3600/16, отображаются данные трендов (даже если история элемента данных за тот же период все еще доступна).
- Если тренды отключены, для построения графика используется история элемента данных, если она доступна за этот период.

Отсутствие данных

Для элементов данных с обычным интервалом обновления на графике ничего не отображается, если данные элемента не собираются.

Однако для элементов данных типа trapper и элементов данных с запланированным интервалом обновления (и обычным интервалом обновления, установленным в 0) проводится прямая линия до первого собранного значения и от последнего собранного значения до конца графика; линия проходит соответственно на уровне первого/последнего значения.

Переключение на необработанные значения

Раскрывающийся список в правом верхнем углу позволяет переключиться с простого графика на списки *Значения/500 последних значений*. Это может быть полезно для просмотра числовых значений, на основе которых построен график.

Представленные здесь значения являются необработанными, то есть для них не используются единицы измерения или постобработка. Однако применяется сопоставление значений.

Известные проблемы

Смотрите известные проблемы по графикам.

Пользовательские графики

Обзор

Пользовательские графики, как следует из названия, предлагают возможность индивидуальной настройки.

В то время как простые графики хороши для просмотра данных по одному элементу данных, они не дают возможности индивидуальной настройки.

Таким образом, если вы захотите изменить стиль графика или способ отрисовки линий, или сравнить несколько элементов данных, например, входящий и исходящий трафик на одном графике, то вам необходим пользовательский график.

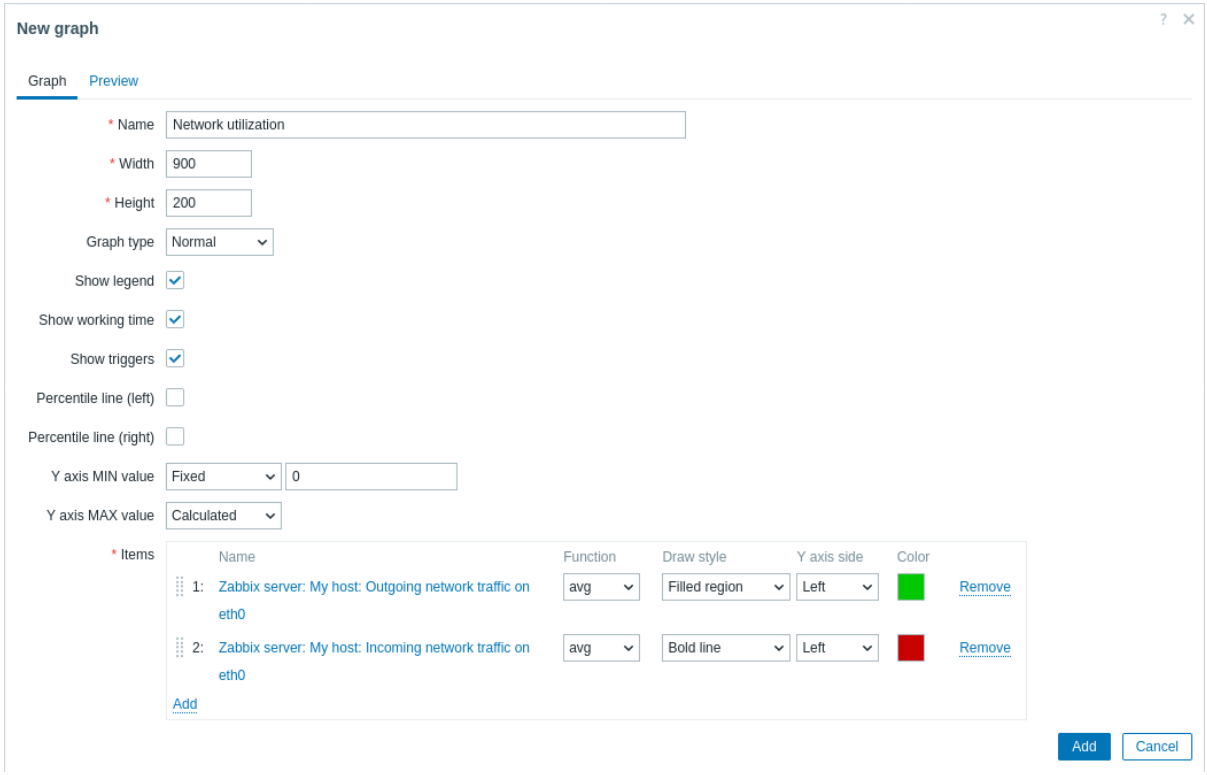
Пользовательские графики настраиваются вручную.

Они могут быть созданы для одного узла сети, для нескольких узлов сети или для отдельного шаблона.

Настройка пользовательских графиков

Чтобы создать пользовательский график, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Узлы сети* (или *Шаблоны*).
2. Нажмите *Графики* в строке рядом с нужным узлом сети или шаблоном.
3. На экране графиков нажмите *Создать график*.
4. Отредактируйте атрибуты графика.



Все обязательные поля ввода отмечены красной звездочкой.

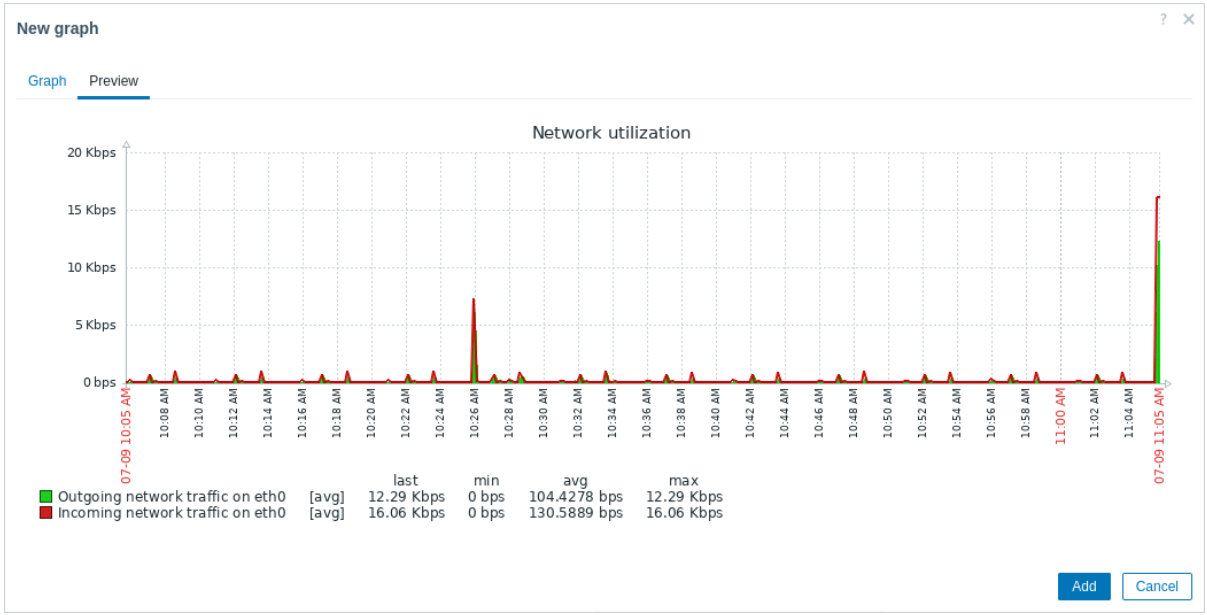
Атрибуты графика:

Параметр	Описание
Имя	Уникальное имя графика. В этом поле поддерживаются выражения макросов, но только с функциями `avg`, `last`, `min` и `max`, с временем в качестве параметра (например, `{?avg(/host/key,1h)}`). Макросы {HOST.HOST<1-9>} поддерживаются для использования внутри этого макроса, ссылаясь на первый, второй, третий и т. д. узел сети в графике, например, `{?avg(/{HOST.HOST2}/key,1h)}`. Обратите внимание, что ссылка на первый узел сети с помощью этого макроса избыточна, так как на первый узел сети можно сослаться неявно, например, `{?avg(//key,1h)}`.
Ширина	Ширина графика в пикселях (только для предварительного просмотра и круговых/разорванных круговых графиков).
Высота	Высота графика в пикселях.
Тип графика	Тип графика: **Normal** - обычный график, значения отображаются в виде линий **Stacked** - накопительный график, отображаются заполненные области **Pie** - круговой график **Exploded** - ”разорванный” круговой график, части отображаются как ”вырезанные” из круга
Показывать легенду	При установке этого флажка будет отображаться легенда графика.
Показывать рабочее время	Если выбрано, нерабочие часы будут отображаться на сером фоне. Этот параметр недоступен для круговых и разорванных круговых графиков.
Показывать триггеры	Если выбрано, простые триггеры будут отображаться в виде линий с черными штрихами поверх цвета важности триггера. Этот параметр недоступен для круговых и разорванных круговых графиков.
Линия перцентиля (левая)	Отображать перцентиль для левой оси Y. Например, если задан перцентиль 95%, линия перцентиля будет находиться на уровне, ниже которого располагаются 95 процентов значений. Отображается ярко-зеленой линией. Доступно только для обычных графиков.
Линия перцентиля (правая)	Отображать перцентиль для правой оси Y. Например, если задан перцентиль 95%, линия перцентиля будет находиться на уровне, ниже которого располагаются 95 процентов значений. Отображается ярко-красной линией. Доступно только для обычных графиков.

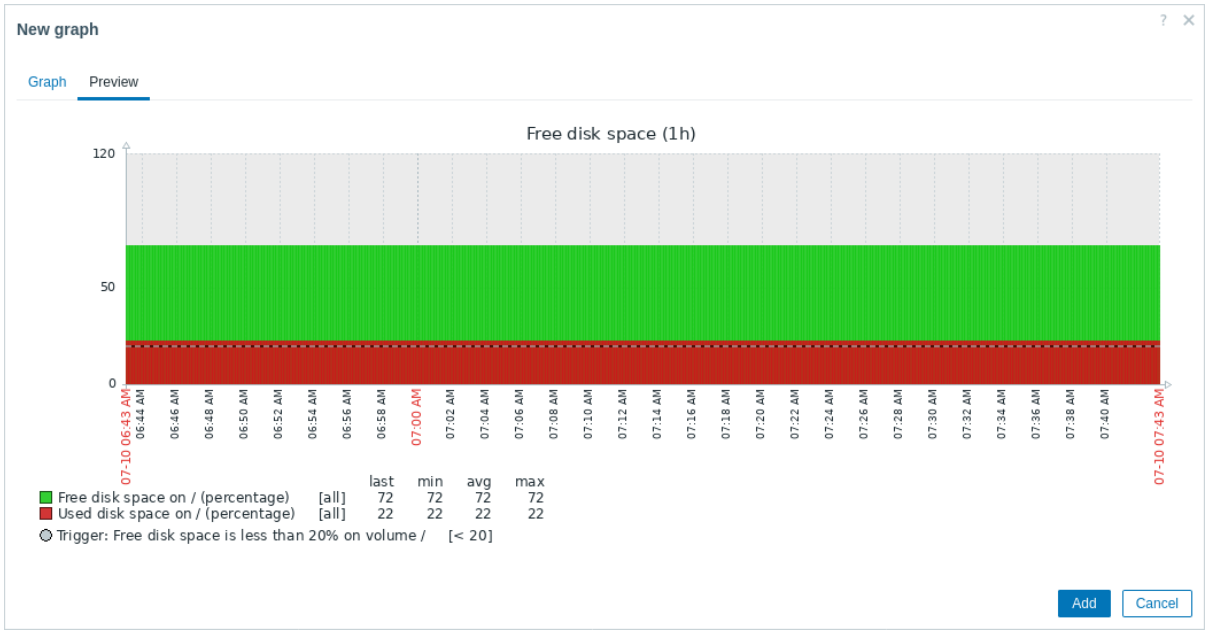
Параметр	Описание
МИН. значение оси Y	Минимальное значение оси Y: **Calculated** - минимальное значение оси Y будет вычислено автоматически. **Fixed** - фиксированное минимальное значение оси Y. **Item** - последнее значение выбранного элемента данных будет минимальным значением. Этот параметр недоступен для круговых и разорванных круговых графиков.
МАКС. значение оси Y	Максимальное значение оси Y: **Calculated** - максимальное значение оси Y будет вычислено автоматически. **Fixed** - фиксированное максимальное значение оси Y. **Item** - последнее значение выбранного элемента данных будет максимальным значением. Этот параметр недоступен для круговых и разорванных круговых графиков.
3D view	Включить стиль 3D. Только для круговых и разорванных круговых графиков.
Элементы данных	Элементы данных, данные которых должны отображаться в этом графике. Нажмите *Добавить*, чтобы выбрать элементы данных. Также можно выбрать различные параметры отображения (функция, стиль рисования, отображение на левой/правой оси, цвет).
Порядок сортировки (0→100)	Порядок отрисовки. 0 будет обработан первым. Можно использовать для отображения линий или областей позади (или перед) другими. Вы можете перетаскивать элементы с помощью значка в начале строки, чтобы задать порядок сортировки или определить, какой элемент отображается перед другим.
Имя	Имя выбранного элемента данных отображается как ссылка. При нажатии на ссылку открывается список других доступных элементов данных.
Тип	Тип (доступно только для круговых и разорванных круговых графиков): **Simple** - значение элемента данных отображается пропорционально на круговой диаграмме **Graph sum** - значение элемента данных представляет весь круговой график Обратите внимание, что цвет элемента ”graph sum” будет виден только в той мере, в какой он не перекрыт элементами ”proportional”.
Функция	Выберите, какие значения будут отображаться, если для элемента данных существует более одного значения на вертикальный пиксель графика: **all** - отображать все возможные значения (минимум, максимум, среднее) на графике. Обратите внимание, что для более коротких периодов этот параметр не влияет на отображение; только для более длительных периодов, когда плотность данных в вертикальном пикселе графика увеличивается, ’all’ начинает отображать минимальные, максимальные и средние значения. Эта функция доступна только для типа графика *Normal*. См. также: Создание графиков из истории/трендов. **avg** - отображать средние значения **last** - отображать последние значения. Эта функция доступна только если в качестве типа графика выбран *Pie/Exploded pie*. **max** - отображать максимальные значения **min** - отображать минимальные значения
Стиль рисования	Выберите стиль рисования (доступно только для обычных графиков; для накопительных графиков всегда используется заполненная область) для применения к данным элемента данных - *Line*, *Bold line*, *Filled region*, *Dot*, *Dashed line*, *Gradient line*.
Сторона оси Y	Выберите сторону оси Y, на которой будут отображаться данные элемента данных - *Left*, *Right*.
Цвет	Выберите цвет для применения к данным элемента данных.

Предпросмотр графика

На вкладке *Предпросмотр (Preview)* отображается предпросмотр графика, так что вы сразу сможете увидеть, что создаёте.



Обратите внимание, что предварительный просмотр не будет отображать какие-либо данные по элементам данных из шаблона.



В этом примере, обратите внимание на жирную пунктирную линию, которая отображает уровень триггера, и на информацию о триггере, которая отображается в легенде.

Note:
Отображается не более 3 линий триггеров. Если триггеров больше, приоритет отображения отдается триггерам с более низкой важностью.

 Если высота графика меньше 120 пикселей, то триггеры в легенде к графику отображаться не будут.

Графики по запросу

Обзор

Простой график удобен для просмотра данных одного элемента данных, а пользовательские графики предлагают возможности настройки, однако ни один из этих вариантов не позволяет быстро создать график сравнения для нескольких элементов данных с минимальными усилиями и затратами на обслуживание.

Для решения этой проблемы можно очень быстро создавать графики по запросу для нескольких элементов данных.

Конфигурация

Чтобы создать ad-hoc-график, выполните следующие действия:

1. Перейдите в *Monitoring* > *Latest data*.
2. Используйте фильтр, чтобы отобразить нужные элементы данных.
3. Отметьте флажками элементы данных, которые нужно вывести на график.
4. Нажмите кнопки *Display stacked graph* или *Display graph*.

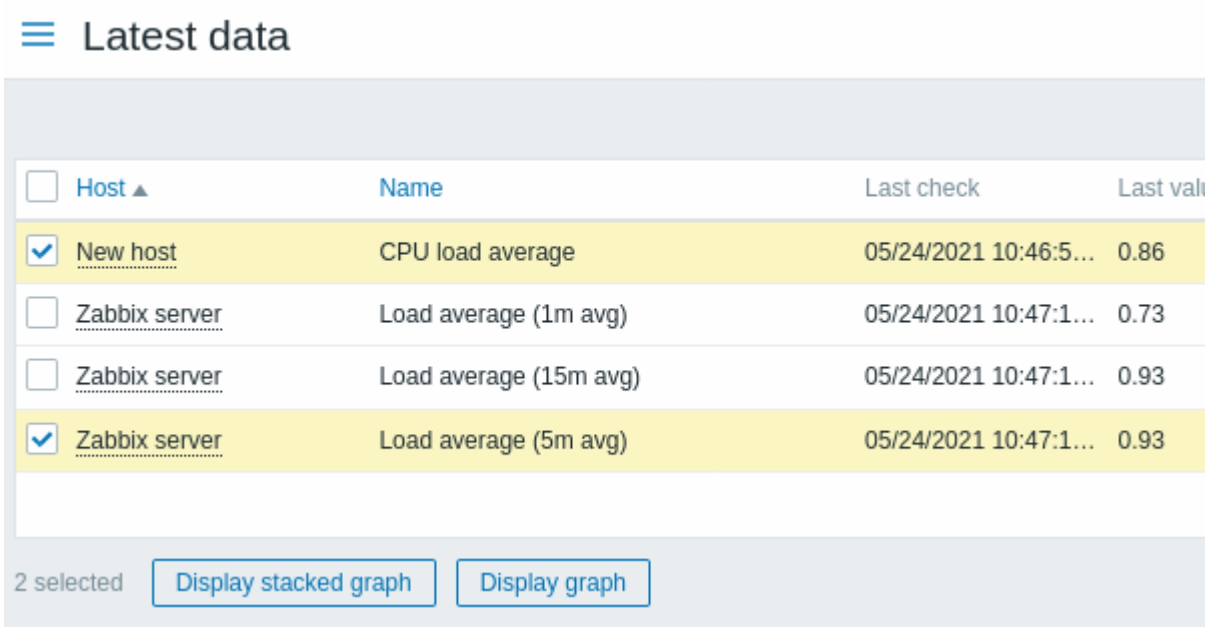
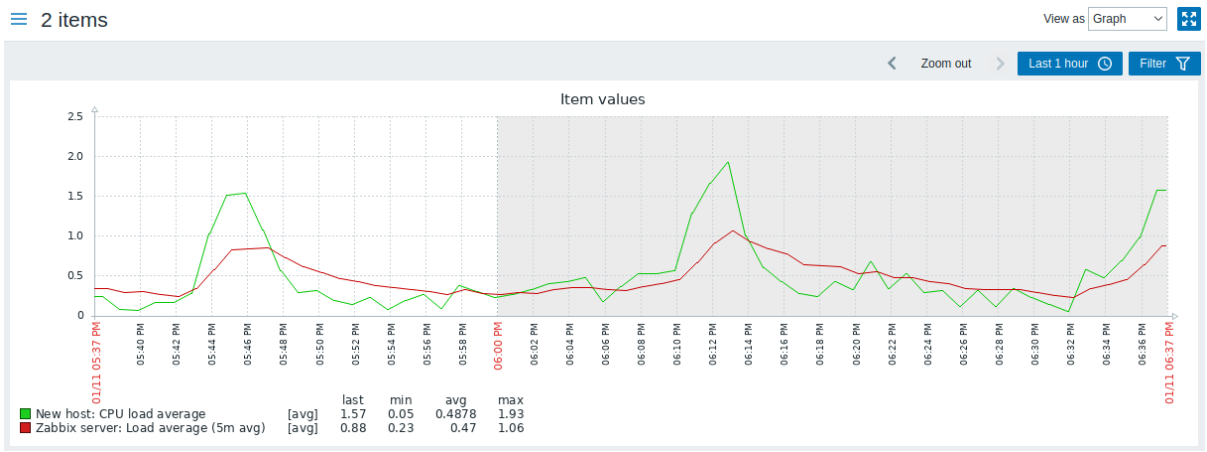
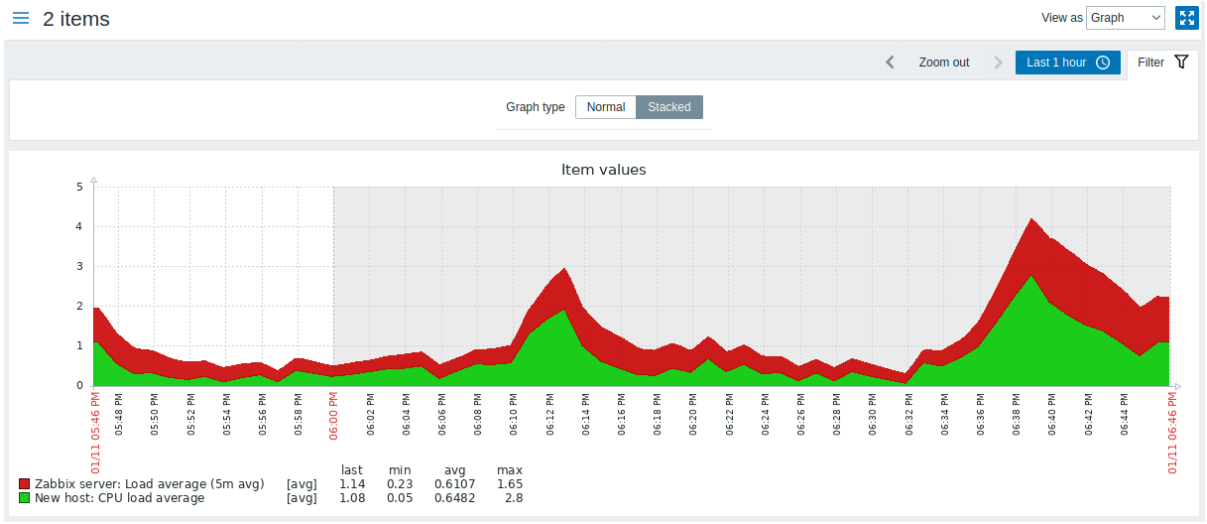


График будет создан мгновенно:



Обратите внимание: чтобы не отображать слишком много линий на графике, для каждого элемента данных показывается только среднее значение (линии минимального/максимального значения не отображаются). Триггеры и информация о триггерах на графике не отображаются.

В окне созданного графика доступен селектор периода времени, а также возможность переключаться с обычного линейного графика на stacked-график и обратно.



Сетевые карты

Обзор

Если у вас есть сеть, за которой вы присматриваете, то вы, возможно, захотите где-либо иметь обзор вашей инфраструктуры. Для этого вы можете создать карты в Zabbix — сетей или всего того, чего пожелаете.

Все пользователи могут создавать карты сети. Карты могут быть публичными (доступными всем пользователям) или приватными (доступными выбранным пользователям).

Перейдём к настройке карты сети.

Настройка карты сети

Обзор

Настройка карты в Zabbix требует, чтобы сначала вы создали карту, задав ее общие параметры, а затем начали заполнять саму карту элементами и их связями.

Вы можете заполнить карту элементами, такими как узел сети, группа узлов сети, триггер, изображение или другая карта.

Для представления элементов карты используются значки. Вы можете определить информацию, которая будет отображаться рядом со значками, и задать особый способ отображения недавних проблем. Вы можете связать значки и определить информацию, которая будет отображаться на связях.

Вы можете добавлять пользовательские URL-адреса, доступные при щелчке по значкам. Таким образом, вы можете связать значок узла сети со свойствами узла сети или значок карты с другой картой.

Количество проблем на картах отображается только для первопричинных проблем.

Карты управляются в *Мониторинг* > *Карты*, где их можно настраивать, управлять ими и просматривать. В представлении мониторинга вы можете щелкать по значкам и использовать ссылки на некоторые скрипты и URL-адреса.

Сетевые карты основаны на векторной графике (SVG).

Публичные и приватные карты

Все пользователи Zabbix (включая пользователей не администраторов) могут создавать карты сети. Карты сети имеют владельца — пользователя, который их создал. Карты сети можно сделать публичными или приватными.

- *Публичные (Public)* карты видны всем пользователям; однако, пользователи должны иметь права на чтение по крайней мере одного элемента карты сети, чтобы её увидеть. Публичные карты можно редактировать, если пользователь / группа пользователей имеет права чтения-записи на эту карту и по крайней мере права на чтение всех элементов на соответствующей карте, включая триггеры в связях.
- *Приватные (Private)* карты видны только своим владельцам и пользователям/группам пользователей, с которыми поделился владелец карты. Обычные пользователи (не Супер-администраторы) могут назначать общий доступ только тем группам, в которые они входят сами, и пользователям этих групп. Пользователи с уровнем Администратор могут видеть приватные карты вне зависимости от того, являются ли они их владельцами или принадлежат ли списку пользователей, с которыми поделились картой. Приватные карты могут редактироваться

владельцами и в случае, если у пользователя/группы пользователей имеются права чтения-записи на карту, а также хотя бы права на чтение всех элементов соответствующей карты, включая триггеры в связях.

Элементы карты, к которым у пользователя отсутствует доступ на чтение, отображаются с серой иконкой, и вся текстовая информация по элементу скрыта. Однако, подпись к триггеру остаётся видимой, даже если у пользователя нет прав на триггер.

Для добавления элемента на карту у пользователя также должны быть права как минимум на чтение этого элемента.

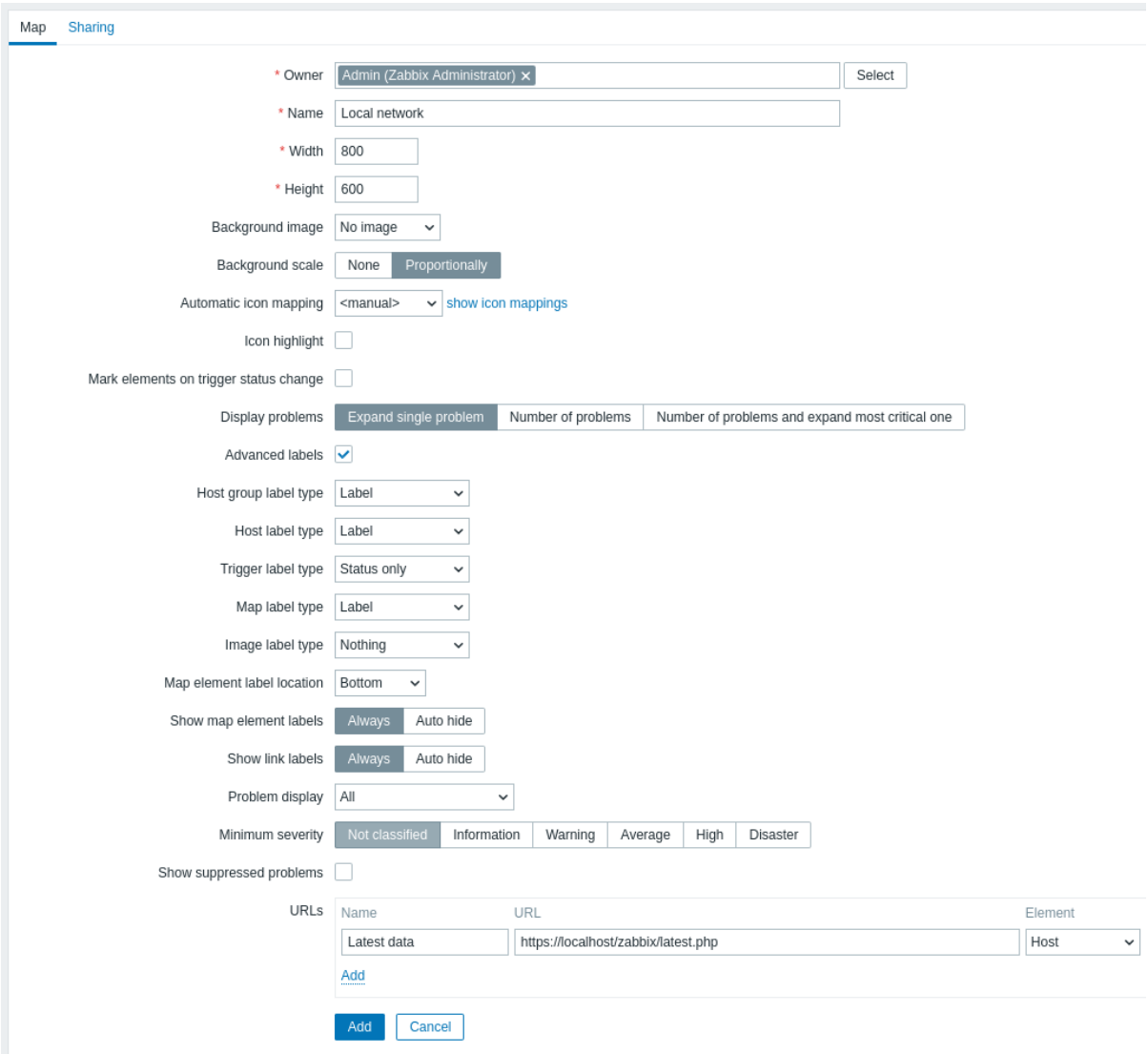
Создание карты

Чтобы создать карту, выполните следующие действия:

1. Перейдите в раздел *Мониторинг* > *Карты*.
2. Перейдите к представлению со всеми картами.
3. Нажмите *Создать карту*.
4. Настройте параметры карты.

Также можно использовать кнопку *Клонировать* в форме настройки существующей карты, чтобы создать новую карту. Эта карта будет иметь все свойства существующей карты, включая общие атрибуты макета, а также элементы существующей карты.

Вкладка **Карта** содержит общие атрибуты карты:



Все обязательные поля ввода отмечены красной звездочкой.

Общие атрибуты карты:

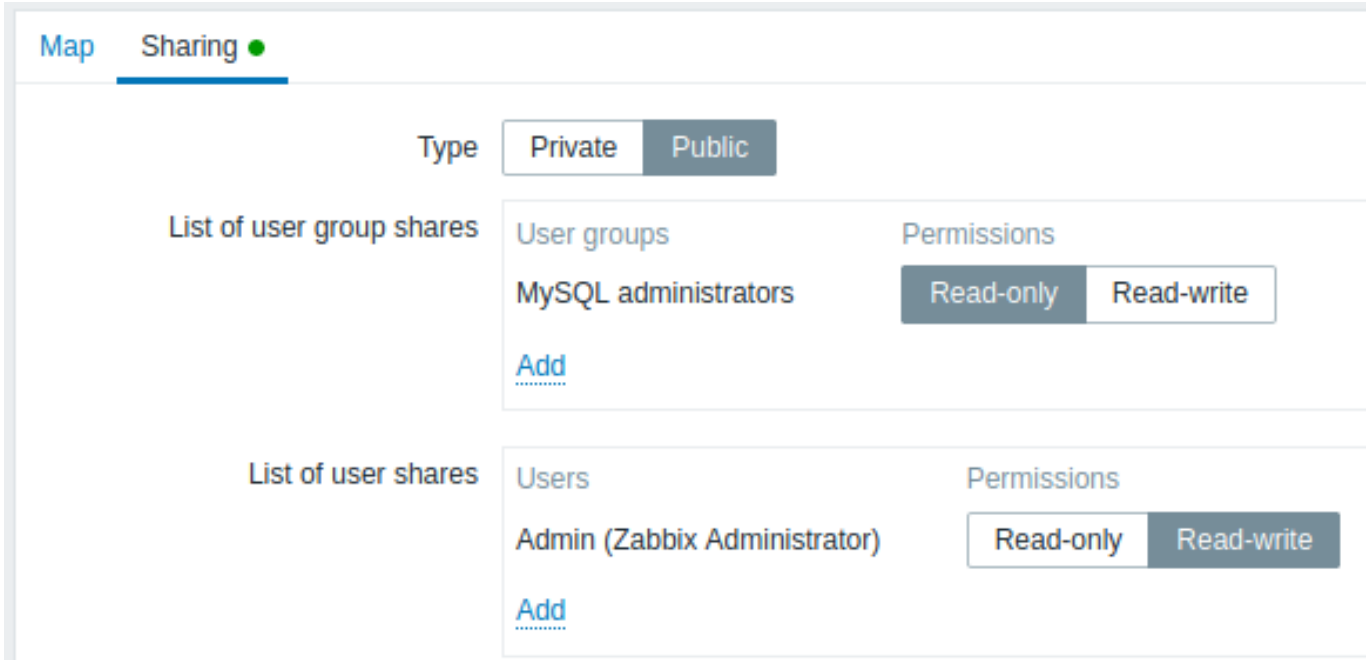
Параметр	Описание
Владелец	Имя владельца карты.
Имя	Уникальное имя карты.
Ширина	Ширина карты в пикселях.

Параметр	Описание
Высота	Высота карты в пикселях.
Фоновое изображение	Использование фонового изображения: **Нет изображения** - фоновое изображение не используется (белый фон) **Изображение** - выбранное изображение используется в качестве фонового. Можно использовать географическую карту или любое другое изображение для улучшения карты.
Масштабирование фона	Использование масштабирования фона: **Нет** - масштабирование не используется; **Пропорционально** - покрыть весь фон карты, сохранив пропорции изображения (по умолчанию).
Автоматическое сопоставление значков	Можно включить автоматическое сопоставление значков, настроенное в разделе *Администрирование > Общее > Сопоставление значков*. Сопоставление значков позволяет сопоставлять определенные значки с определенными полями инвентаризации узла сети.
Выделение значков	Если установить этот флажок, элементы карты будут выделяться. Элементы с активным триггером получат круглую область фона того же цвета, что и триггер с наивысшим уровнем серьезности. Кроме того, вокруг круга будет отображаться толстая зеленая линия, если все проблемы подтверждены. Элементы со статусом «отключен» или «на обслуживании» получат квадратную область фона серого и оранжевого цвета соответственно. См. также: Просмотр карт
Отмечать элементы при изменении статуса триггера	Недавнее изменение состояния триггера (недавняя проблема или восстановление) будет выделено маркерами (красными треугольниками, направленными внутрь) на трех сторонах значка элемента, свободных от подписи. Маркеры отображаются в течение 30 минут.
Отображение проблем	Выберите способ отображения проблем с элементом карты: **Развернуть одну проблему** - если существует только одна проблема, отображается ее имя. В противном случае отображается общее количество проблем. **Количество проблем** - отображается общее количество проблем **Количество проблем и развернуть наиболее критичную** - отображаются имя наиболее критичной проблемы и общее количество проблем. «Наиболее критичная» определяется на основе серьезности проблемы, а при равной серьезности - идентификатора события проблемы (сначала отображается проблема с более высоким идентификатором или более поздняя проблема). Для *элемента карты триггеров* учитываются серьезность проблемы и, при равной серьезности, позиция триггера в списке триггеров. Если существует несколько проблем одного триггера, отображается самая последняя.
Расширенные подписи	Если установить этот флажок, можно будет определить отдельные типы подписей для разных типов элементов.
Тип подписи группы узлов сети	Тип подписи, используемый для элемента карты: **Подпись** - подпись элемента карты **IP-адрес** - IP-адрес **Имя элемента** - имя элемента (например, имя узла сети) **Только статус** - только статус (OK или ПРОБЛЕМА) **Ничего** - подписи не отображаются
Тип подписи узла сети *Тип подписи триггера* *Тип подписи карты* *Тип подписи изображения*	
Расположение подписи элемента карты	Расположение подписи относительно элемента карты: **Снизу** - под элементом карты **Слева** - слева **Справа** - справа **Сверху** - над элементом карты
Показывать подписи элементов карты	Выберите способ отображения подписей элементов карты: **Всегда** - отображаются всегда (по умолчанию); **Скрывать автоматически** - скрывать подпись, если на нее не наведена мышь и она не выбрана.

Параметр	Описание
Показывать подписи связей	Выберите способ отображения подписей связей: **Всегда** - отображаются всегда (по умолчанию); **Скрывать автоматически** - скрывать подпись, если на нее не наведена мышь и она не выбрана.
Отображение проблем	Отображать количество проблем как: **Все** - отображается полное количество проблем **Раздельно** - количество неподтвержденных проблем отображается отдельно как часть общего количества проблем **Только неподтвержденные** - отображается только количество неподтвержденных проблем
Минимальная серьезность	Проблемы ниже выбранного минимального уровня серьезности не будут отображаться на карте. Например, если выбран уровень *Предупреждение*, изменения срабатываний с уровнями *Информация* и *Не классифицировано* не будут отражаться на карте.
Показывать подавленные проблемы	Установите этот флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не отображались) из-за обслуживания узла сети.
URL	Здесь можно определить URL для каждого типа элементов, где: **Имя** - подпись URL; **URL** - URL (до 2048 символов); **Элемент** - тип элемента (например, *Узел сети*, *Группа узлов сети* и т. д.). Они будут отображаться как ссылки, когда пользователь нажимает на элемент в режиме просмотра карты. В именах и значениях URL карты можно использовать макросы. Полный список см. в разделе поддерживаемые макросы, выполнив поиск по фразе «имена и значения URL карты».

Общий доступ

Вкладка **Общий доступ** содержит тип карты, а также параметры общего доступа (группы пользователей, пользователи) для приватных карт:



Parameter	Description
Type	Выберите тип карты: **Private** - карта видна только выбранным группам пользователей и пользователям **Public** - карта видна всем
List of user group shares	Выберите группы пользователей, которым доступна карта. Можно разрешить доступ только для чтения или для чтения и записи.

Parameter	Description
List of user shares	Выберите пользователей, которым доступна карта. Можно разрешить доступ только для чтения или для чтения и записи.

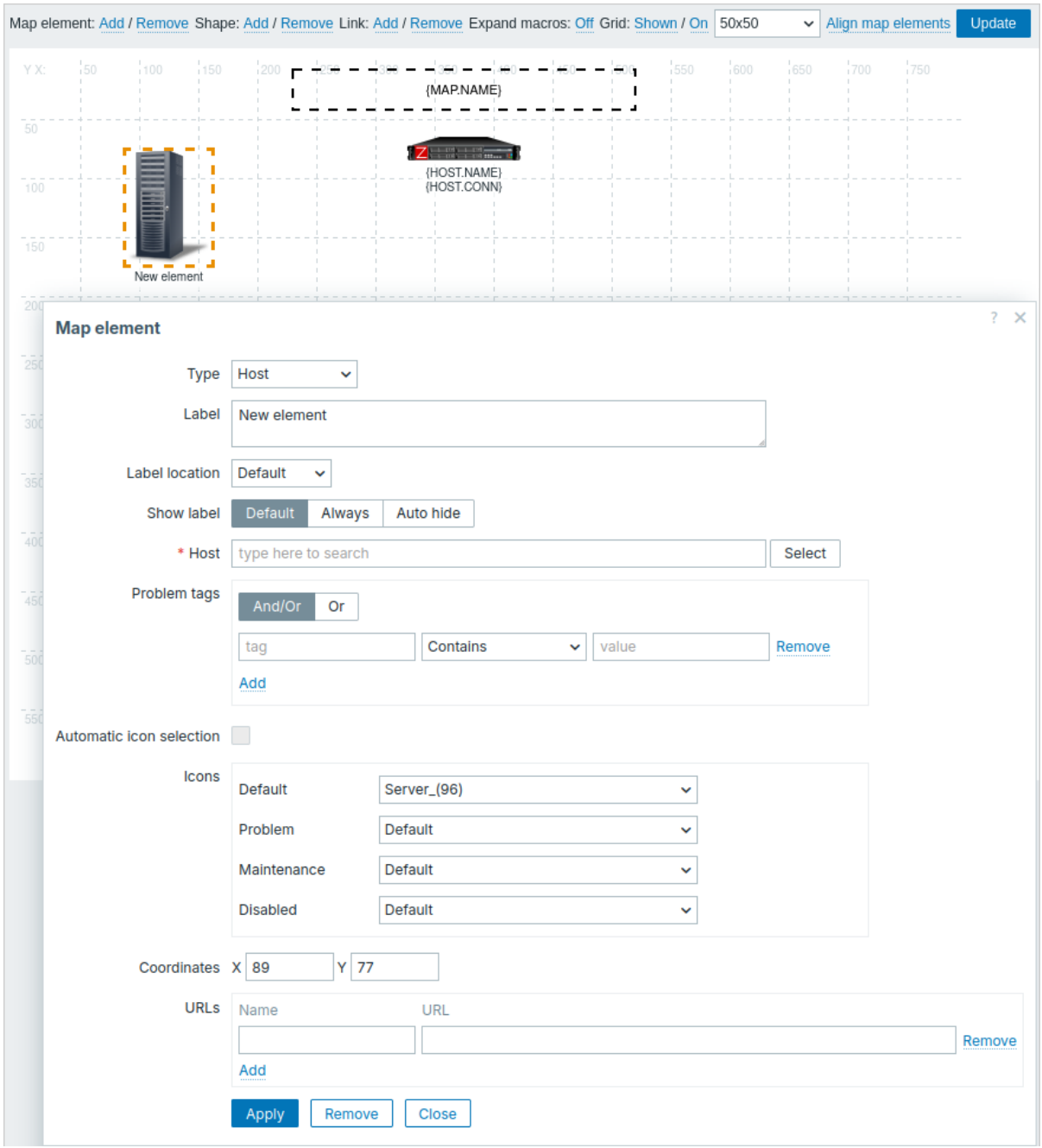
Когда вы нажимаете *Add* для сохранения этой карты, создается пустая карта с именем, размерами и определенными настройками. Теперь нужно добавить несколько элементов. Для этого нажмите *Edit* в списке карт, чтобы открыть область редактирования.

Добавление элементов

Чтобы добавить элемент, нажмите *Add* рядом с `Map element`. Новый элемент появится в левом верхнем углу карты. Перетащите его в любое удобное место.

Обратите внимание: если для параметра Grid включено значение "On", элементы всегда будут выравниваться по сетке (можно выбрать разные размеры сетки в выпадающем списке, а также скрыть/показать сетку). Если вы хотите размещать элементы где угодно без выравнивания, переключите параметр в "Off". (Позже можно выровнять случайно размещенные элементы по сетке, нажав *Align map elements*.)

Теперь, когда у вас уже есть несколько элементов, вы, возможно, захотите начать различать их, задавая им имена и т. д. При нажатии на элемент отображается форма, в которой можно задать тип элемента, указать имя, выбрать другой значок и т. д.



Атрибуты элемента карты:

Parameter	Description
Type	Тип элемента: **Host** - значок, отображающий состояние всех триггеров выбранного узла сети **Map** - значок, отображающий состояние всех элементов карты **Trigger** - значок, отображающий состояние одного или нескольких триггеров **Host group** - значок, отображающий состояние всех триггеров всех узлов сети, входящих в выбранную группу **Image** - значок, не связанный ни с каким ресурсом
Label	Подпись значка, любая строка. Можно использовать макросы и многострочные строки. В этом поле поддерживаются макросы выражений macros, но только с функциями `avg`, `last`, `min` и `max`, с временем в качестве параметра (например, `{?avg(/host/key,1h)}`). Полный список поддерживаемых макросов см. в разделе supported macros и найдите 'map element labels'.
Label location	Расположение подписи относительно значка: **Default** - расположение подписи по умолчанию для карты; **Bottom** - под значком; **Left** - слева; **Right** - справа; **Top** - над значком.
Show label	Выберите, как отображать подпись элемента: **Default** - использовать настройки *Show map element labels* из конфигурации карты (по умолчанию); **Always** - всегда отображать (по умолчанию); **Auto hide** - скрывать подпись, если на нее не наведены или она не выбрана.
Host	Выберите узел сети; либо введите имя узла сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет показан выпадающий список подходящих узлов сети. Прокрутите вниз, чтобы выбрать один из них. Нажмите значок удаления (x), чтобы снять выбор. Это поле доступно только для элементов типа *Host*.
Map	Выберите карту; либо введите имя карты. Это поле поддерживает автодополнение, поэтому при вводе имени карты будет показан выпадающий список подходящих карт. Прокрутите вниз, чтобы выбрать одну из них. Нажмите значок удаления (x), чтобы снять выбор. Это поле доступно только для элементов типа *Map*.
Triggers	Выберите один или несколько триггеров в поле *New triggers* ниже и нажмите *Add*. Либо начните вводить имя триггера и выберите один из выпадающего списка подходящих триггеров. Нажмите значок удаления (x), чтобы снять выбор. Это поле доступно только для элементов типа *Trigger*. Порядок выбранных триггеров можно изменить, но только в пределах одной и той же серьезности триггеров. Выбор нескольких триггеров также влияет на разрешение макросов {HOST.*} как в режиме редактирования, так и в режиме просмотра. В режиме редактирования первые отображаемые макросы {HOST.*} будут разрешаться в зависимости от первого триггера в списке (на основе серьезности триггера). Режим просмотра зависит от параметра Display problems в общих атрибутах карты: - Если выбран режим *Expand single problem*, первые отображаемые макросы {HOST.*} будут разрешаться в зависимости от последнего обнаруженного проблемного триггера (независимо от серьезности) или первого триггера в списке (если проблем не обнаружено). - Если выбран режим *Number of problems and expand most critical one*, первые отображаемые макросы {HOST.*} будут разрешаться в зависимости от серьезности триггера.
Host group	Выберите группу узлов сети; либо введите группу узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы узлов сети будет показан выпадающий список подходящих групп узлов сети. Прокрутите вниз, чтобы выбрать одну из них. Нажмите значок удаления (x), чтобы снять выбор. Это поле доступно только для элементов типа *Host group*.

Parameter	Description
Problem tags	Укажите теги, чтобы ограничить число проблем, отображаемых в виджете. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия Это поле доступно только для элементов типа *Host* и *Host group*.
Automatic icon selection	Если флажок установлен, для определения отображаемого значка будет использоваться сопоставление значков.
Icons	Можно выбрать разные значки для элемента в следующих случаях: *Default*, *Problem*, *Maintenance* и *Disabled*. Это поле доступно только для элементов типа *Host*.
Coordinates	Координаты элемента карты по осям **X** и **Y**.
URLs	Здесь для элемента можно задать URL, специфичные для элемента, где: **Name** - подпись для URL; **URL** - URL (до 2048 символов). Они будут отображаться как ссылки, когда пользователь нажимает на элемент в режиме просмотра карты. Если у элемента есть собственные URL и для его типа определены URL на уровне карты, они будут объединены в одном меню. В именах и значениях элементов карты можно использовать макросы. Полный список см. в разделе supported macros и найдите 'map URL names and values'.

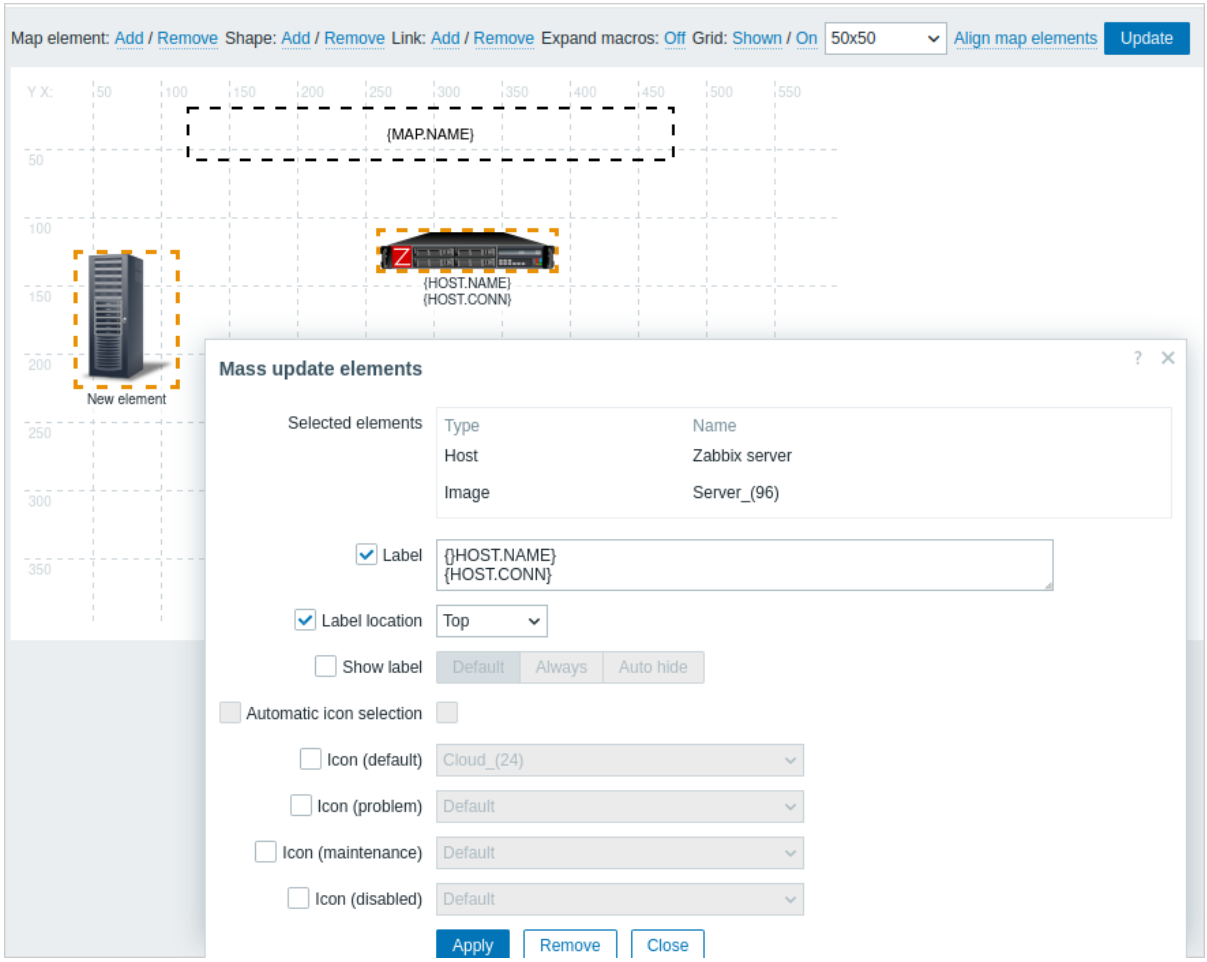
Attention:
Добавленные элементы не сохраняются автоматически. Если вы перейдете на другую страницу, все изменения могут быть потеряны.
Поэтому рекомендуется нажать кнопку **Update** в правом верхнем углу. После нажатия изменения сохраняются независимо от того, что вы выберете во всплывающем окне ниже.
Выбранные параметры сетки также сохраняются для каждой карты.

Выбор элементов

Для выбора нескольких элементов, выберите сначала один элемент, затем нажмите и удерживайте *Ctrl*, чтобы выбрать другие.

Вы также можете выбрать несколько элементов, растягивая прямоугольник в области редактирования и тем самым выбирая все элементы, находящиеся в нём.

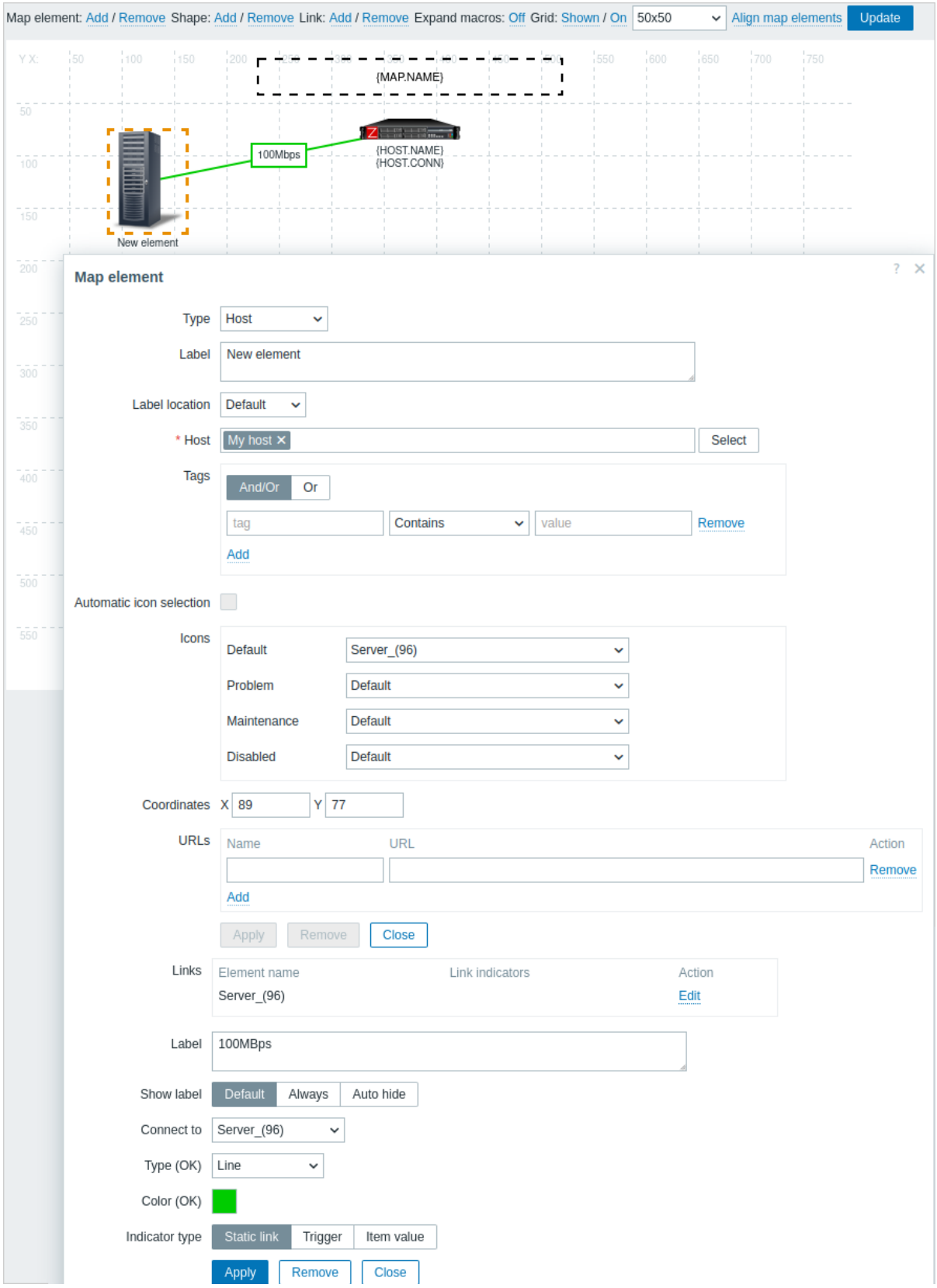
Как только вы выберете более одного элемента, диалог свойств элемента переключится в режим массового обновления, так что вы сможете изменить атрибуты выбранных элементов за раз. Чтобы это сделать, отметьте атрибут, используя флажок, и укажите для него новое значение. Здесь вы можете использовать макросы (такие как, скажем, {HOST.NAME} для подписи элемента).



Связывание элементов

После того как вы разместили на карте несколько элементов, можно приступать к их связыванию. Чтобы связать два элемента, сначала нужно выделить их. После выделения элементов нажмите *Добавить* рядом с Link.

После создания связи в форме отдельного элемента появляется дополнительный раздел *Links*. Нажмите *Редактировать*, чтобы изменить атрибуты связи.



Атрибуты связи:

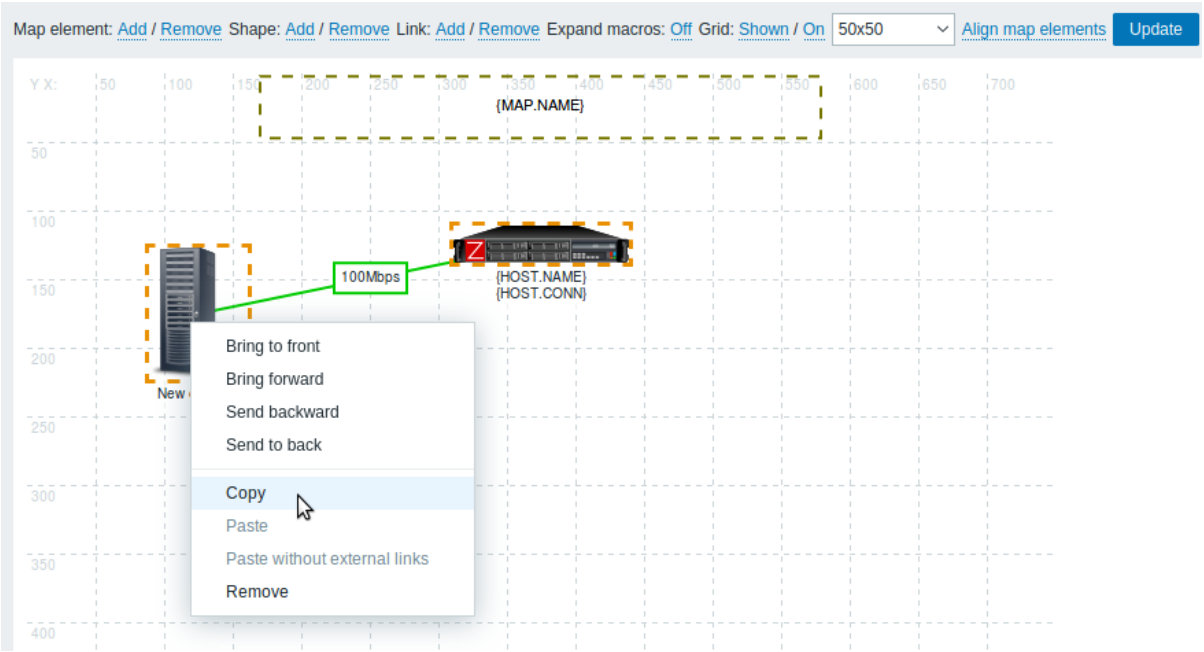
Parameter	Description
Label	Метка, которая будет отображаться над связью. В этом поле поддерживаются выражения macros, но только с функциями `avg`, `last`, `min` и `max`, с временем в качестве параметра (например, `{?avg(/host/key,1h)}`).

Parameter	Description
Show label	Выберите, как отображать метку связи: **Default** - использовать настройки *Show map element labels* из конфигурации карты (по умолчанию); **Always** - всегда отображать (по умолчанию); **Auto hide** - скрывать метку, если на нее не наведено указатель мыши и она не выделена.
Connect to	Элемент, к которому подключается связь.
Type (OK)	Стиль связи по умолчанию. Выберите стиль: *Line*, *Bold line*, *Dot* или *Dashed line*.
Color (OK)	Цвет связи по умолчанию. Нажмите на образец цвета, чтобы выбрать другой цвет.
Indicator type	Выберите тип индикатора связи: **Static link** - без индикаторов; **Trigger** - разрешить индикаторы связи на основе триггеров; **Item value** - разрешить индикаторы связи на основе значений элементов данных.
Item	Выберите элемент данных. Значение этого элемента данных будет влиять на стиль связи. Это поле доступно, когда для *Indicator type* выбрано *Item value*.
Indicators	Список link indicators. Это поле доступно, когда для *Indicator type* установлено значение *Trigger* или *Item value*. Если выбрано *Trigger*, в качестве индикаторов связи можно назначать триггеры. Когда триггер находится в состоянии *Problem*, к связи будет применен его стиль (выбранный цвет и тип линии). Если выбрано *Item value* и в поле *Item* выбран элемент данных, в качестве индикаторов связи можно назначать значения элементов данных. Когда значение элемента данных достигает указанного порога (для числового элемента данных) или совпадает с шаблоном (для текстового элемента данных), к связи применяется его стиль (выбранный цвет и тип линии).

Перемещение и копирование элементов

Несколько выбранных элементов можно **переместить** в другое место на карте, щелкнув по одному из выбранных элементов, удерживая нажатой кнопку мыши и перемещая курсор в нужное место.

Один или несколько элементов можно **скопировать**, выделив элементы, а затем щелкнув по выбранному элементу правой кнопкой мыши и выбрав *Copy* в меню.

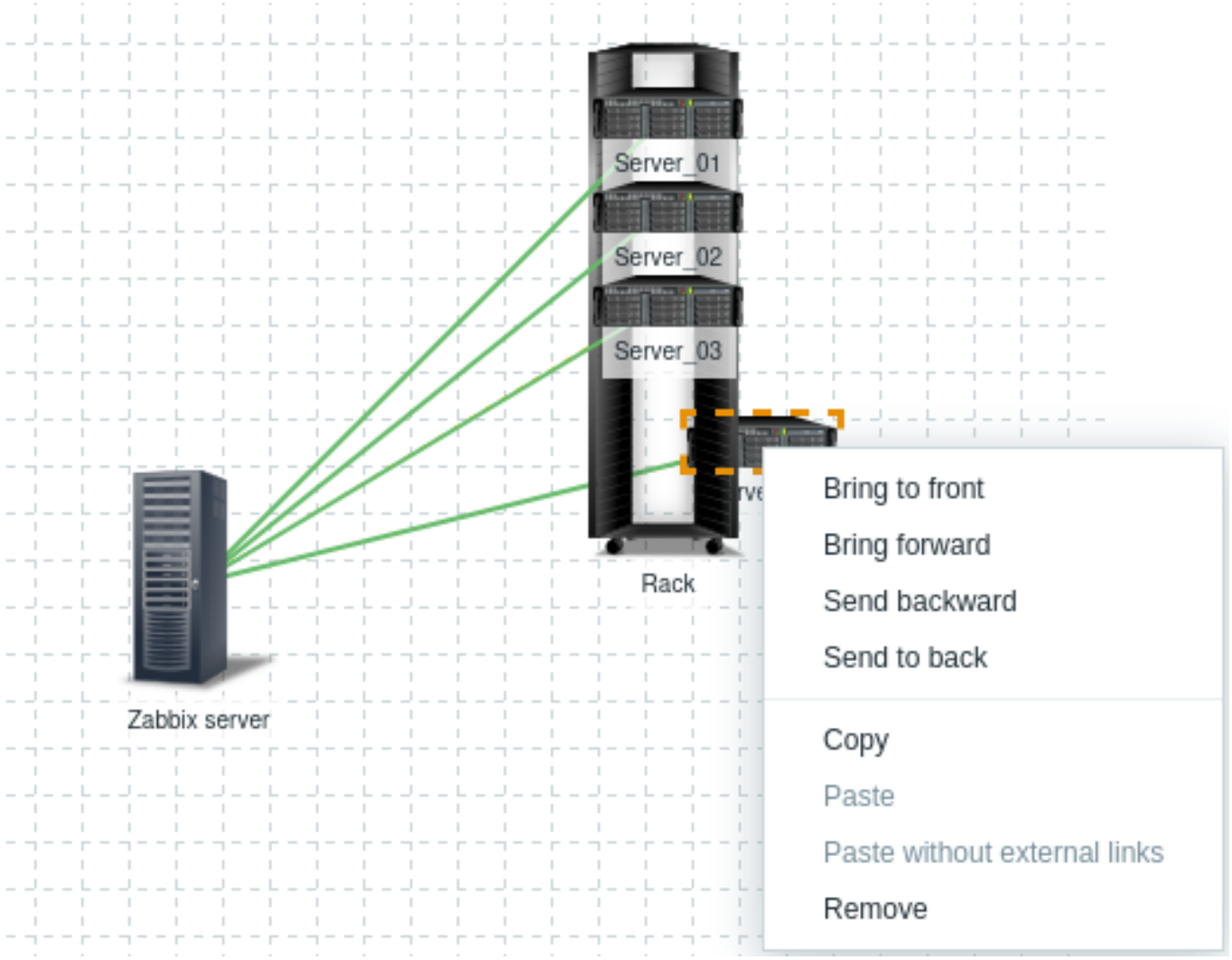


Чтобы вставить элементы, щелкните правой кнопкой мыши по области карты и выберите *Paste* в меню. Параметр *Paste without external links* вставит элементы, сохранив только связи между выбранными элементами.

Копирование и вставка работают в пределах одного окна браузера. Сочетания клавиш не поддерживаются.

Порядок элементов

Чтобы переместить один элемент перед другим (или наоборот), щелкните элемент правой кнопкой мыши и выберите *Переместить вперед/На передний план* или *Переместить назад/На задний план*.

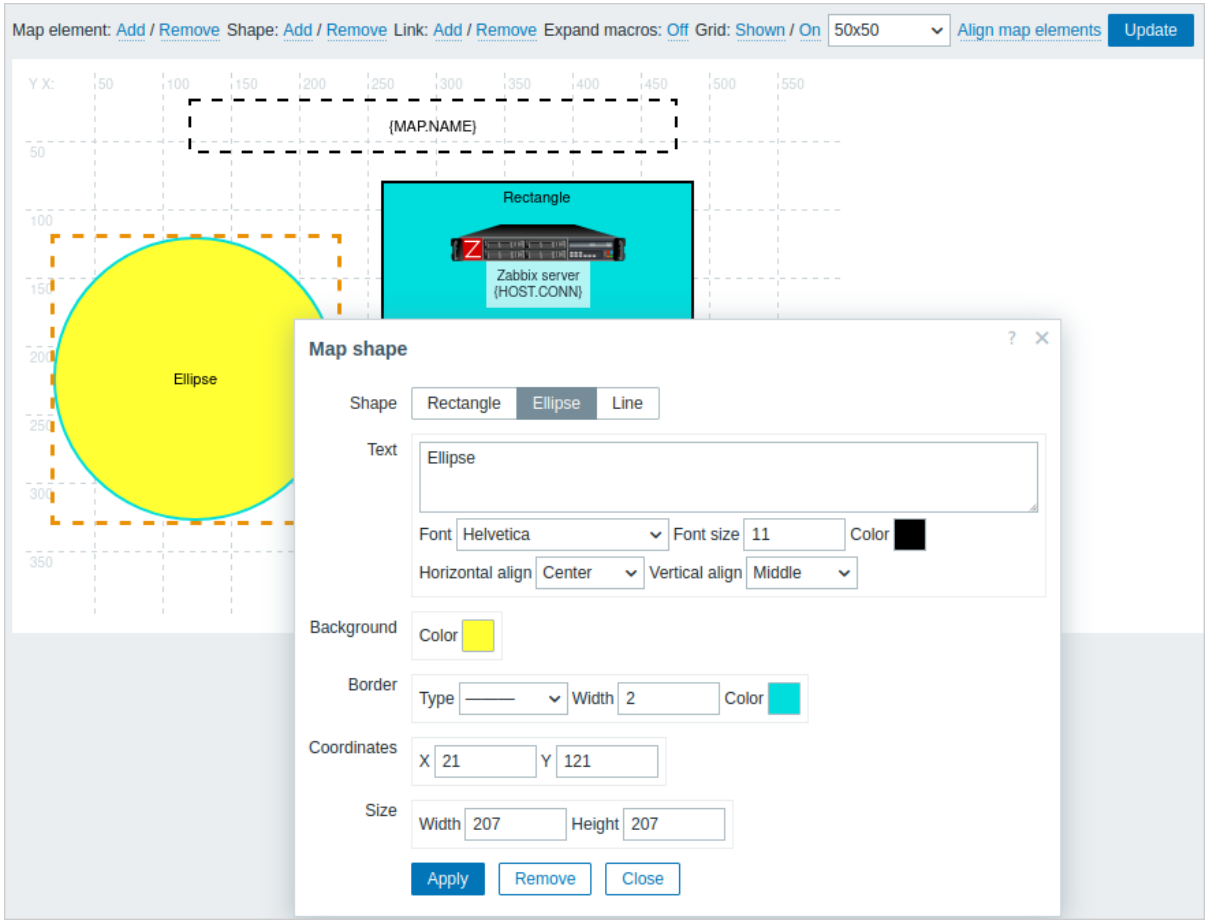


Добавление фигур

Помимо элементов карты, можно также добавлять некоторые фигуры. Фигуры не являются элементами карты; это лишь визуальное представление. Например, фигуру в виде прямоугольника можно использовать в качестве фона для группировки некоторых узлов сети. Можно добавлять фигуры в виде прямоугольника и эллипса.

Чтобы добавить фигуру, нажмите *Добавить* рядом с пунктом Shape. Новая фигура появится в левом верхнем углу карты. Перетащите ее в любое нужное место.

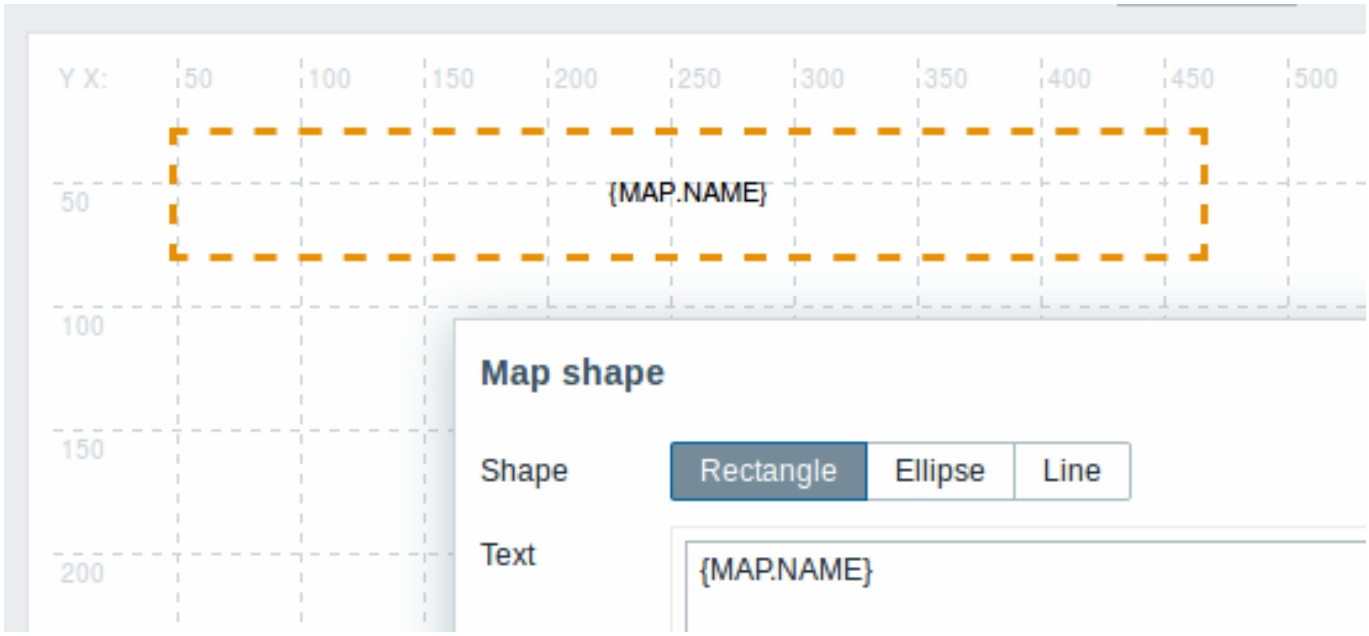
Новая фигура добавляется со стандартными цветами. При нажатии на фигуру отображается форма, в которой можно настроить внешний вид фигуры, добавить текст и т. д.



Чтобы выбрать фигуры, выберите одну из них, а затем, удерживая *Ctrl*, выберите остальные. При выборе нескольких фигур общие свойства можно массово обновлять так же, как и для элементов.

В фигуры можно добавлять текст. В тексте поддерживаются выражения макросов, но только с функциями `avg`, `last`, `min` и `max`, с временем в качестве параметра (например, `{?avg(/host/key,1h)}`).

Чтобы отображать только текст, фигуру можно сделать невидимой, удалив ее границу (выберите 'None' в поле *Border*). Например, обратите внимание, что макрос {MAP.NAME}, видимый на скриншоте выше, на самом деле является прямоугольной фигурой с текстом, который можно увидеть при нажатии на макрос:



{MAP.NAME} подставляет настроенное имя карты при просмотре карты.

Если в тексте используются гиперссылки, при просмотре карты по ним можно будет перейти.

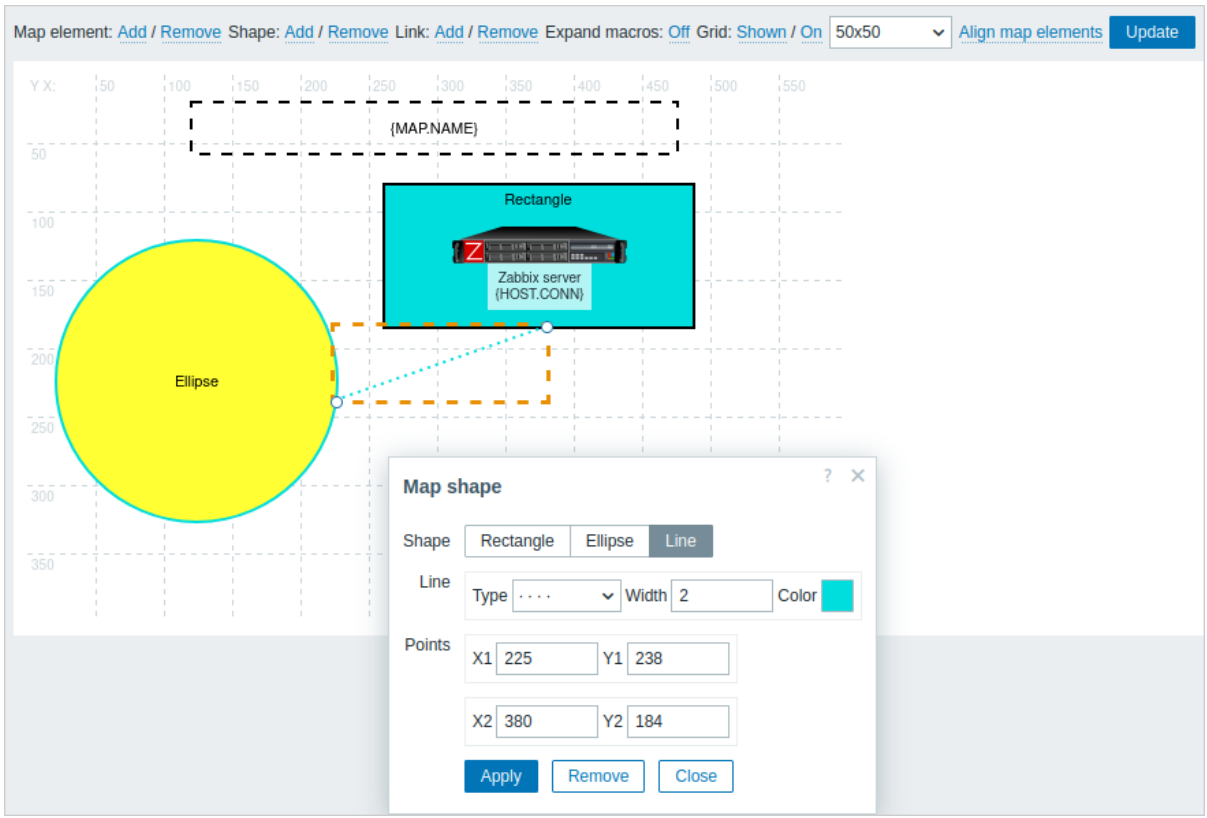
Перенос строк в тексте внутри фигур всегда включен. Однако внутри эллипса строки переносятся так, как если бы эллипс был прямоугольником. Перенос по словам не реализован, поэтому длинные слова (слова, которые не помещаются

в фигуру) не переносятся, а маскируются (на странице редактирования карты) или обрезаются (на других страницах с картами).

Добавление линий

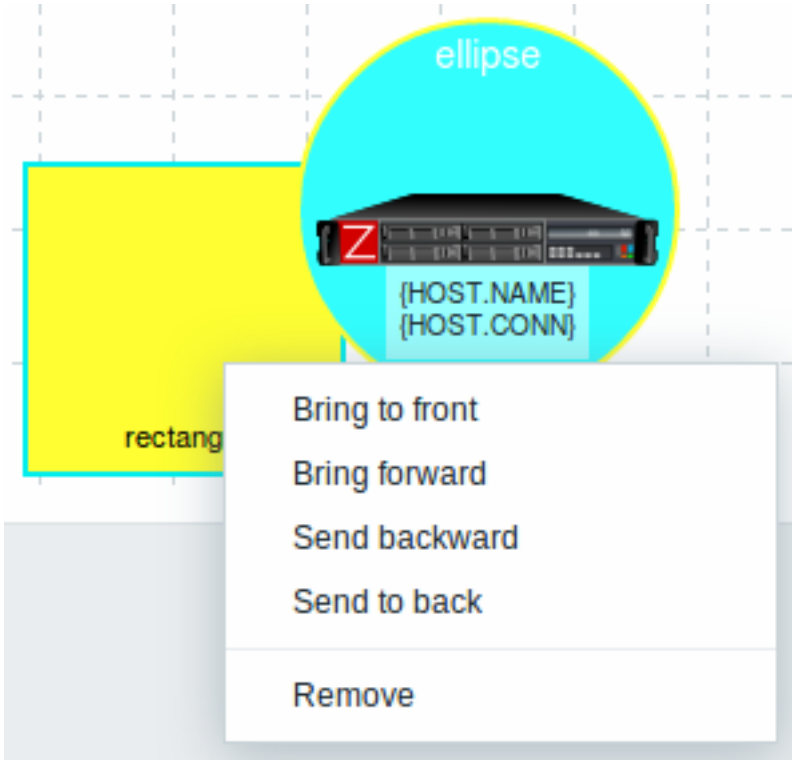
Помимо фигур, можно также добавлять линии. Линии можно использовать для связывания элементов или фигур на карте.

Чтобы добавить линию, нажмите *Добавить* рядом с Shape. Новая фигура появится в левом верхнем углу карты. Выберите ее и нажмите *Line* в форме редактирования, чтобы изменить фигуру на линию. Затем настройте свойства линии, такие как тип линии, ширина, цвет и т. д.



Упорядочивание фигур и линий

Чтобы переместить одну фигуру перед другой (или наоборот), щелкните фигуру правой кнопкой мыши, чтобы открыть меню фигуры карты.

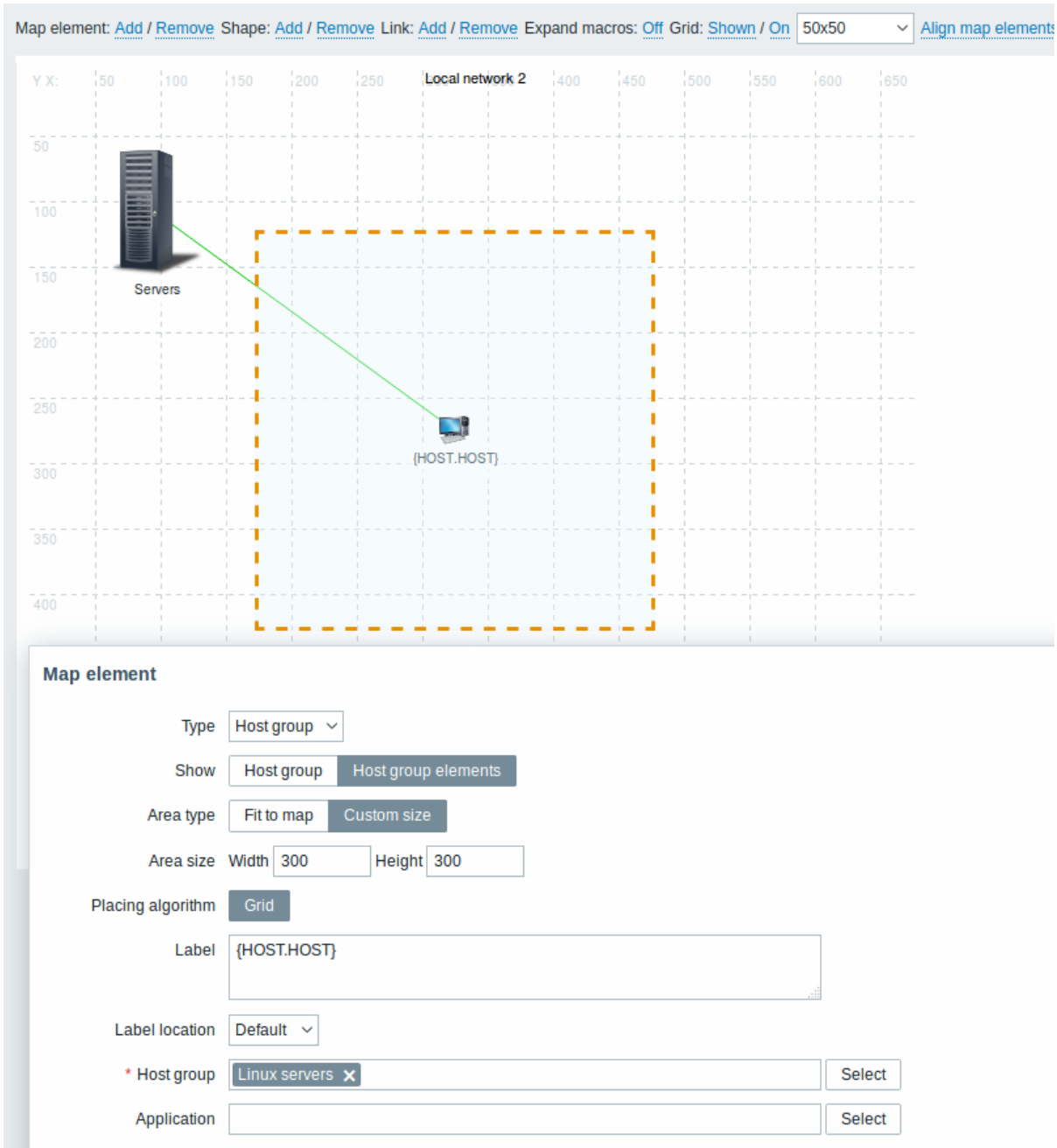


Элементы группы узлов сети

Обзор

Этот раздел объясняет, как добавить элемент типа *Группа узлов сети* при настройке карты сети.

Конфигурация



Все обязательные поля ввода отмечены красной звездочкой.

Эта таблица содержит параметры, характерные для типа элемента *Группа узлов сети*:

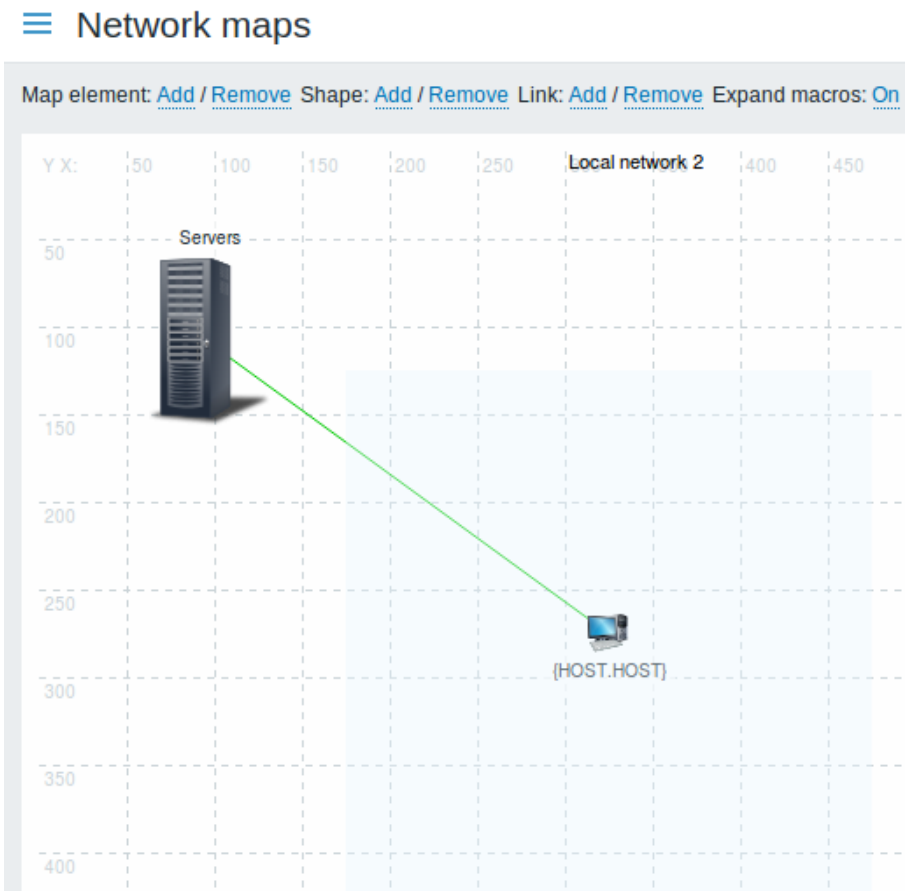
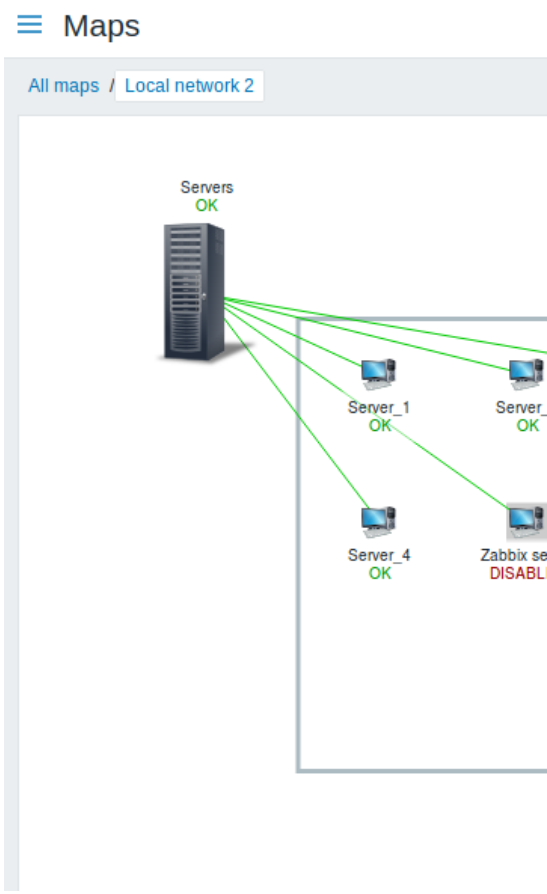
Параметр	Описание
Тип	Выберите тип элемента: **Группа узлов сети** — значок, отображающий состояние всех триггеров всех узлов сети, входящих в выбранную группу.
Отображать	Варианты отображения: **Группа узлов сети** — выбор этого варианта приведет к отображению одного значка с соответствующей информацией об определенной группе узлов сети. **Элементы группы узлов сети** — выбор этого варианта приведет к отображению нескольких значков с соответствующей информацией о каждом отдельном элементе (узле сети) определенной группы узлов сети.

Параметр	Описание
Тип области	Этот параметр доступен, если для поля *Отображать* установлено значение *Элементы группы узлов сети*: **Подогнать под карту** — все элементы группы узлов сети равномерно размещаются в пределах карты. **Пользовательский размер** — ручная настройка области карты для отображения всех элементов группы узлов сети.
Размер области	Этот параметр доступен, если для поля *Тип области* установлено значение *Пользовательский размер*: **Ширина** — значение в пикселях, задающее ширину области карты. **Высота** — значение в пикселях, задающее высоту области карты.
Алгоритм размещения	**Сетка** — единственный доступный вариант отображения всех элементов группы узлов сети.
Подпись	Подпись значка, любая строка. В подписях можно использовать макросы и многострочные строки. Если тип элемента карты — *Группа узлов сети*, указание определенных макросов влияет на представление карты, отображая соответствующую информацию о каждом отдельном узле сети. Например, если используется макрос `{HOST.IP}`, в режиме редактирования карты будет отображаться только сам макрос `{HOST.IP}`, а в режиме просмотра карты будут отображаться уникальные IP-адреса каждого узла сети.

Просмотр элементов группы узлов сети

Этот параметр доступен, если выбран параметр отображения *Элементы группы узлов сети*.

При выборе *Элементы группы узлов сети* в качестве параметра *отображения* сначала отображается только один значок группы узлов сети. Однако после сохранения карты и перехода к просмотру карты вы увидите, что карта содержит все элементы (узлы сети) определенной группы узлов сети:

Представление редактирования карты	Представление карты
	

Обратите внимание, как используется макрос `{HOST.HOST}`. В режиме редактирования карты имя макроса не разрешается, а в режиме просмотра карты отображаются уникальные имена всех узлов сети.

Индикаторы ссылок

Обзор

Вы можете назначать индикаторы ссылке между элементами на карте сети.

Индикаторы могут основываться на триггерах или значениях элементов данных. Можно отображать различный стиль и цвет ссылки:

- когда триггеры переходят в состояние проблемы;
- когда значение элемента данных:
  - достигает порогового значения (для числовых элементов данных);
  - соответствует регулярному выражению (для текстовых элементов данных).

При настройке ссылки вы задаете тип и цвет ссылки по умолчанию. Назначая индикаторы ссылке, вы можете сделать так, чтобы стиль и цвет ссылки зависели от состояния триггера или значения элемента данных.

Например, если любой из назначенных триггеров перейдет в состояние проблемы, стиль и цвет ссылки изменятся соответствующим образом. Допустим, по умолчанию ваша ссылка была зеленой линией. Теперь, когда триггер находится в состоянии проблемы, ссылка может стать жирной красной линией (если вы так ее настроили).

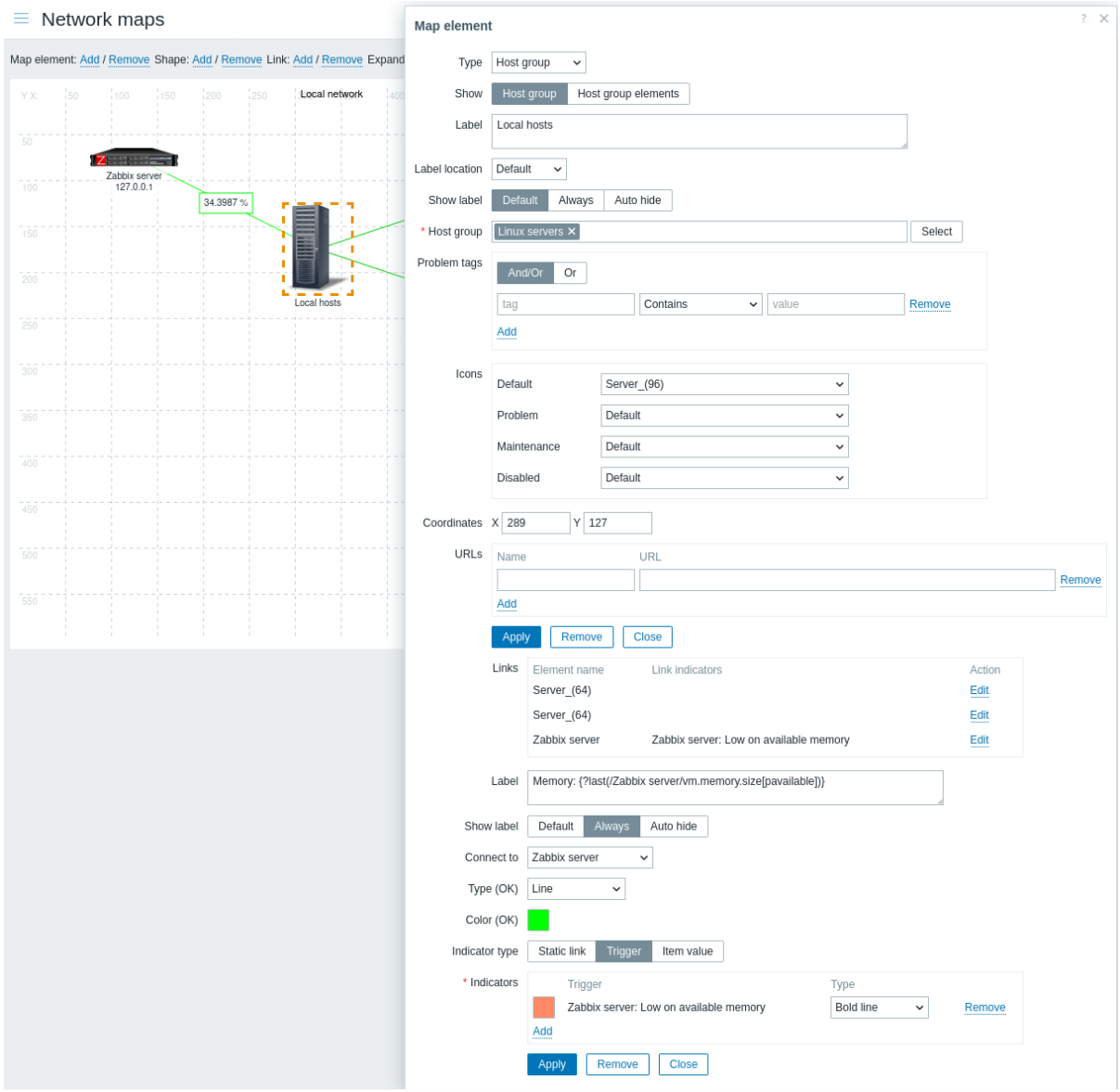
Аналогично, если значение элемента данных достигает заданного порога или соответствует заданному регулярному выражению, стиль ссылки может измениться соответствующим образом.

Конфигурация

Триггеры

Чтобы назначить триггеры в качестве индикаторов связи, выполните следующие действия:

- выберите элемент карты
- нажмите *Edit* в разделе *Links* для соответствующей связи
- выберите *Trigger* в качестве типа индикатора
- нажмите *Add* в блоке *Indicators* и выберите один или несколько триггеров

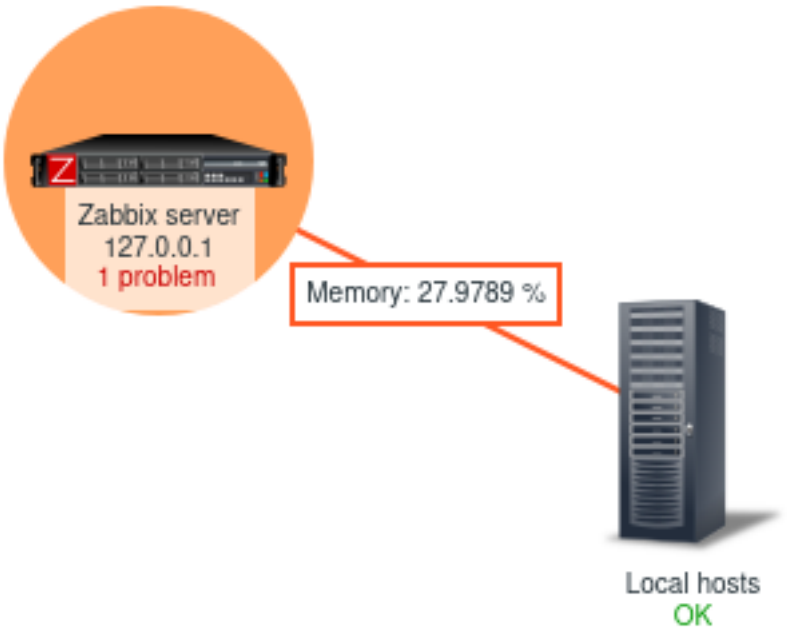


Все обязательные поля ввода отмечены красной звездочкой.

Добавленные триггеры можно увидеть в списке *Indicators*.

Вы можете задать тип связи и цвет для каждого индикатора прямо из списка. После этого нажмите *Apply*, закройте форму и нажмите *Update* , чтобы сохранить изменения карты.

В *Monitoring* > *Maps* соответствующий тип связи и цвет будут отображаться, если **триггер** перейдет в состояние проблемы.



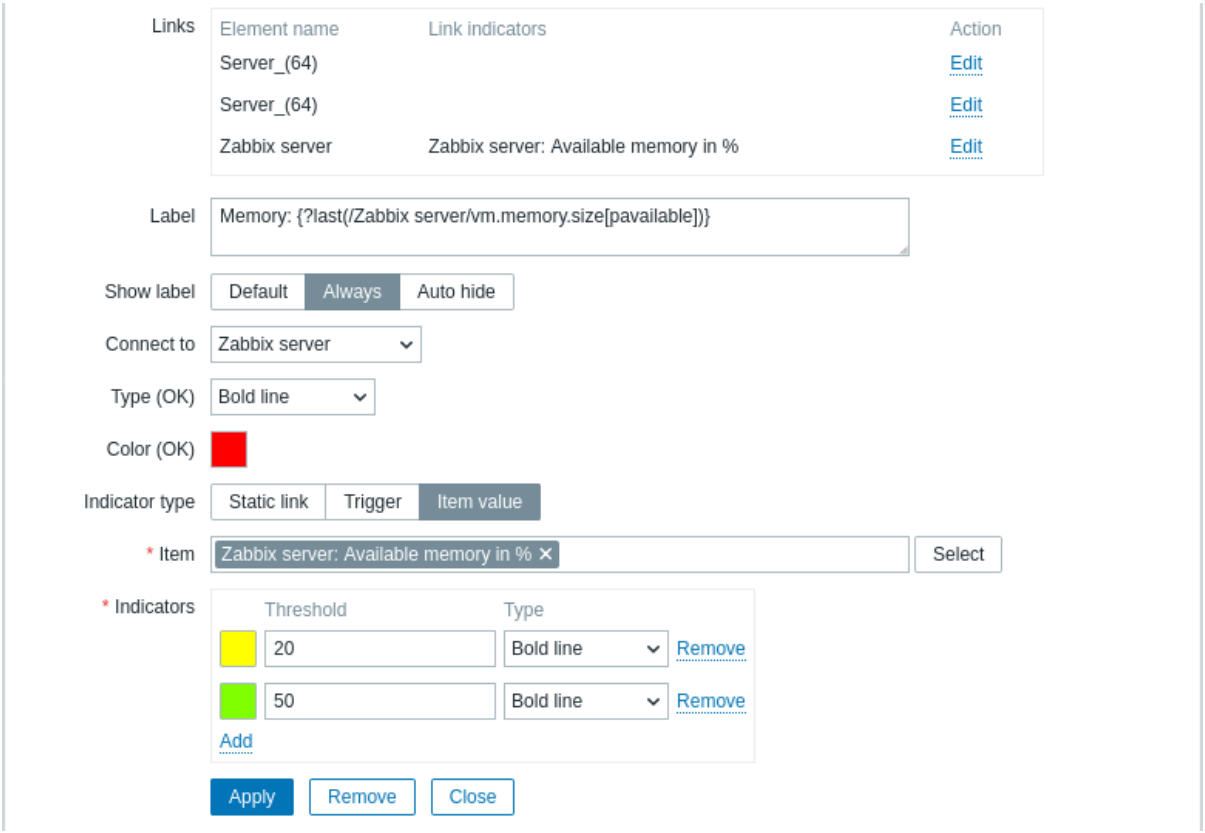
Note:
Если несколько триггеров перейдут в состояние проблемы, то для определения стиля и цвета связи будет использована проблема с наивысшей серьезностью. Если к одной и той же связи карты назначено несколько триггеров с одинаковой серьезностью, приоритет будет у того, у которого наименьший ID. Также обратите внимание:

1. Параметры *Minimum trigger severity* и *Show suppressed problem* в конфигурации карты влияют на то, какие проблемы будут учитываться.
2. В случае триггеров с несколькими проблемами (multiple problem generation) каждая проблема может иметь серьезность, отличающуюся от серьезности триггера (измененной вручную), может иметь разные теги (из-за макросов) и может быть подавлена.

Значения элементов данных

Чтобы назначить значения элементов данных в качестве индикаторов связи, выполните следующие действия:

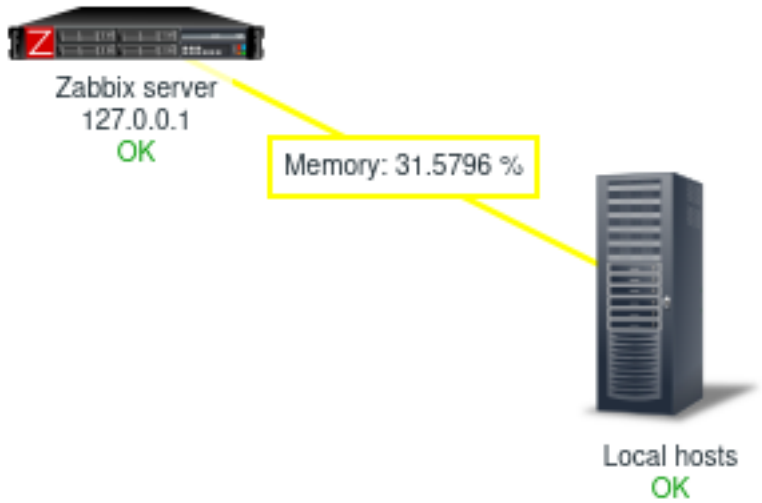
- выберите элемент карты
- нажмите *Edit* в разделе *Links* для соответствующей связи
- выберите *Item value* в качестве типа индикатора
- выберите элемент данных
- добавьте один или несколько пороговых значений или шаблонов значений элемента данных в блоке *Indicators*



Добавленные пороговые значения/шаблоны элементов данных можно увидеть в списке *Indicators*.

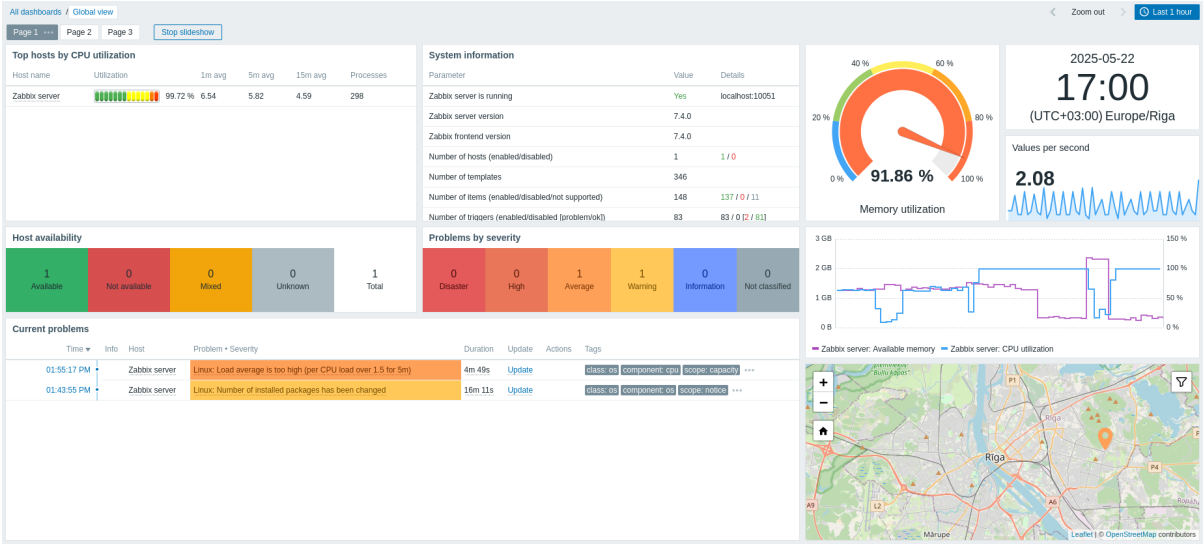
Вы можете задать тип связи и цвет для каждого индикатора прямо из списка. После этого нажмите *Apply*, закройте форму и нажмите *Update* , чтобы сохранить изменения карты.

В *Monitoring* > *Maps* соответствующий тип связи и цвет будут отображаться, если **значение элемента данных** достигает заданного порога (для числовых типов данных) или соответствует шаблону регулярного выражения (для текстовых типов данных).



Панели мониторинга

Панели мониторинга — как глобальные панели мониторинга, так и панели мониторинга узла сети — предоставляют мощную платформу визуализации с такими виджетами и инструментами, как современные графики, карты, слайд-шоу и многое другое.



Шаблоны и группы шаблонов

Обзор

Использование шаблонов — отличный способ сократить рабочую нагрузку и оптимизировать настройки Zabbix. Шаблон — это набор сущностей, которые можно удобно применить к множеству узлов сети.

Сущностями могут быть:

- элементы данных
- триггеры
- графики
- панели
- правила низкоуровневого обнаружения
- веб-сценарии

Поскольку в реальной жизни многие узлы сети идентичны или довольно схожи, естественно следует, что набор сущностей (элементы данных, триггеры, графики...) созданные для одного узла сети, могут быть полезны для других узлов сети. Конечно можно было бы скопировать их на каждый новый узел сети, но это потребовало бы большого количества ручной работы. Вместо этого можно скопировать все такие сущности в один шаблон, а затем применить его к необходимому набору узлов сети.

Когда шаблон присоединён к узлу сети, все сущности (элементы, триггеры, графики,...) шаблона добавляются к узлу сети. Шаблоны назначаются каждому отдельному узлу сети напрямую (а не группе узлов сети).

Шаблоны часто используются для группировки сущностей в для определённых служб или приложений (таких как Apache, MySQL, PostgreSQL, Postfix...) и затем применяются к узлам сети, на которых работают эти службы.

Еще одно преимущество использования шаблонов — когда что-то нужно изменить для всех узлов сети. Изменение чего-либо на уровне шаблона один раз распространяется на все присоединённые к этому шаблону узлы сети.

Шаблоны организованы в группы шаблонов.

Перейдём к созданию и настройке шаблона.

Шаблоны из коробки

Обзор

Zabbix предоставляет постоянно расширяющийся набор предварительно настроенных шаблонов, чтобы упростить и ускорить настройку объектов мониторинга.

Все готовые шаблоны доступны в разделе *Сбор данных* > *Шаблоны*.

При обновлении Zabbix шаблоны не обновляются автоматически. Это позволяет избежать перезаписи внесенных вами пользовательских изменений. Чтобы обновить или добавить шаблоны вручную, см. раздел *Обновление шаблонов* ниже.

Для получения информации о конкретных типах шаблонов и требованиях к их работе используйте боковую панель.

Глобальные панели мониторинга, предназначенные для работы с существующими шаблонами Zabbix, можно скачать из репозитория панелей мониторинга и импортировать в вашу среду Zabbix. Подробное описание каждой панели мониторинга, включая предварительные требования, доступно в файле README этой панели.

См. также:

- Связывание шаблона
- Известные проблемы с шаблонами

Обновление шаблонов

Чтобы обновить или добавить **один шаблон**:

1. Откройте каталог `templates` в Git-репозитории Zabbix.

2. Выберите ветку (под меткой *Source*), соответствующую вашей версии Zabbix (например, *master* для версии в разработке).

3. Найдите шаблон, который хотите добавить или обновить (например, MySQL by Zabbix agent).

4. Скопируйте содержимое файла шаблона (например, `template_db_mysql_agent.yaml`) и сохраните его в локальный файл.

5. В веб-интерфейсе Zabbix перейдите в раздел *Сбор данных* > *Шаблоны* и импортируйте локальный файл.

> **Note:**
> Типы носителей вебхуков можно обновить аналогичным образом: получите их из каталога `media` и импортируйте в разделе *Оповещения* > *Типы носителей*.

Чтобы обновить или добавить **несколько шаблонов**:

1. Скачайте архив с пакетами шаблонов в зависимости от вашего веб-сервера:

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. Распакуйте архив. В нем находятся YAML-файлы, каждый из которых содержит несколько шаблонов, а также индексный файл со списком шаблонов в каждом YAML-файле.

3. В веб-интерфейсе Zabbix перейдите в раздел *Сбор данных* > *Шаблоны* и импортируйте YAML-файлы по одному.

> **Note:**
> Типы носителей вебхуков можно обновить аналогичным образом: возьмите файл `media_bundle-8.0-000.yaml` из архива с пакетами шаблонов и импортируйте его в разделе *Оповещения* > *Типы носителей*.

Чтобы обновить или добавить **все шаблоны** одновременно, можно использовать скрипт `import_templates.sh`, включенный в архив с пакетами шаблонов. Для работы скрипта требуются утилиты `curl` и `jq`; скрипт использует метод API Zabbix `configuration.import`. При запуске скрипта укажите каталог, содержащий YAML-файлы, а также URL API Zabbix и токен аутентификации.

> **Warning:**
> Скрипт `import_templates.sh` перезаписывает все существующие шаблоны без возможности просмотреть или отменить изменения шаблонов.

Пример:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Совместимость шаблонов с мастером узлов сети

Мастер узлов сети совместим со всеми готовыми шаблонами.

Вы также можете сделать собственные шаблоны совместимыми с мастером узлов сети:

1. В веб-интерфейсе Zabbix перейдите в раздел *Сбор данных* > *Шаблоны* и экспортируйте собственный шаблон.

2. Отредактируйте экспортированный файл:

- Добавьте элемент шаблона `wizard_ready`, чтобы указать совместимость с мастером узлов сети.
- При необходимости добавьте элемент шаблона `readme` или `config`.

3. В веб-интерфейсе Zabbix перейдите в раздел *Сбор данных* > *Шаблоны* и импортируйте обновленный файл шаблона.

Настройка шаблонов для работы с агентом Zabbix

Выполните следующие действия, чтобы обеспечить корректную работу шаблонов, собирающих метрики с помощью агента Zabbix:

1. Убедитесь, что агент Zabbix установлен на узле сети. Для активных проверок также убедитесь, что адрес сервера/прокси Zabbix добавлен в параметр `ServerActive` файла конфигурации агента.
2. Свяжите шаблон с целевым узлом сети (если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать его).
3. При необходимости измените значения макросов шаблона.
4. Настройте отслеживаемый экземпляр, чтобы разрешить совместное использование данных с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (его можно открыть, нажав имя шаблона).

Доступны следующие шаблоны:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Работа шаблона Zabbix agent 2

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики с помощью Zabbix agent 2:

1. Убедитесь, что агент 2 установлен на узле сети и что установленная версия содержит необходимый плагин. В некоторых случаях сначала может потребоваться обновить агент 2.
2. Привяжите шаблон к целевому узлу сети (если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать шаблон).
3. При необходимости настройте значения макросов шаблона. Обратите внимание, что пользовательские макросы можно использовать для переопределения параметров конфигурации.
4. Настройте контролируемый экземпляр так, чтобы он разрешал обмен данными с Zabbix.

Attention:
Шаблоны Zabbix agent 2 работают совместно с плагинами. Хотя базовую настройку можно выполнить, просто изменив пользовательские макросы, более глубокую настройку можно выполнить, настроив плагин напрямую. Например, если плагин поддерживает именованные сеансы, можно отслеживать несколько сущностей одного типа (например, MySQL1 и MySQL2), указав для каждой сущности в файле конфигурации именованный сеанс с собственными URI, именем пользователя и паролем.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (открывается при нажатии на имя шаблона).

Доступны следующие шаблоны:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

Операции шаблонов HTTP

Выполните следующие действия, чтобы обеспечить корректную работу шаблонов, собирающих метрики с помощью HTTP-агента:

1. Создайте в Zabbix узел сети, соответствующий цели мониторинга. Укажите IP-адрес или DNS-имя цели мониторинга в качестве основного интерфейса узла сети. Это необходимо для корректного разрешения макроса `{HOST.CONN}` в элементах данных шаблона.
2. Свяжите шаблон с узлом сети, созданным на шаге 1 (если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать его).
3. При необходимости измените значения макросов уровня шаблона.
4. Настройте мониторируемый экземпляр так, чтобы он разрешал обмен данными с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле `README` шаблона (его можно открыть, нажав имя шаблона).

Доступны следующие шаблоны:

- Acronis Cyber Protect Cloud по HTTP
- Apache по HTTP
- Asterisk по HTTP
- AWS по HTTP
- AWS Cost Explorer по HTTP
- AWS EC2 по HTTP
- AWS ECS Cluster по HTTP
- AWS ECS Serverless Cluster по HTTP
- AWS ELB Application Load Balancer по HTTP
- AWS ELB Network Load Balancer по HTTP
- AWS Lambda по HTTP
- AWS RDS instance по HTTP
- AWS S3 bucket по HTTP
- Azure по HTTP
- Cisco Meraki organization по HTTP
- Cisco SD-WAN по HTTP
- Cisco Secure Firewall Threat Defense по HTTP
- Claude API по HTTP
- ClickHouse по HTTP
- Cloudflare по HTTP
- CockroachDB по HTTP
- Control-M enterprise manager по HTTP
- Control-M server по HTTP
- Cradlepoint NCM v2 по HTTP
- DELL PowerEdge R720 по HTTP
- DELL PowerEdge R740 по HTTP

- DELL PowerEdge R820 по HTTP
- DELL PowerEdge R840 по HTTP
- Domain RDAP по HTTP
- Elasticsearch Cluster по HTTP
- Envoy Proxy по HTTP
- Etcd по HTTP
- FortiGate по HTTP
- GitHub repository по HTTP
- GitLab по HTTP
- GLPI по HTTP
- Google Cloud Platform (GCP) по HTTP
- Hadoop по HTTP
- HAProxy по HTTP
- HashiCorp Consul Cluster по HTTP
- HashiCorp Consul Node по HTTP
- HashiCorp Nomad по HTTP
- HashiCorp Vault по HTTP
- Hikvision camera по HTTP
- HPE iLO по HTTP
- HPE MSA 2040 Storage по HTTP
- HPE MSA 2060 Storage по HTTP
- HPE Primera по HTTP
- HPE Synergy по HTTP
- InfluxDB по HTTP
- Jenkins по HTTP
- Kubernetes API server по HTTP, Kubernetes cluster state по HTTP, Kubernetes Controller manager по HTTP, Kubernetes kubelet по HTTP, Kubernetes nodes по HTTP, Kubernetes Scheduler по HTTP являются устаревшими и заменены шаблонами *Kubernetes Cluster по HTTP* и *Kubernetes Node по HTTP*.
- Kubernetes Cluster по HTTP
- Kubernetes Node по HTTP
- MantisBT по HTTP
- Microsoft 365 reports по HTTP
- Microsoft SharePoint по HTTP
- NetApp AFF A700 по HTTP
- Nextcloud по HTTP
- NGINX по HTTP
- NGINX Plus по HTTP
- Nutanix Prism Element по HTTP
- OpenAI Platform по HTTP
- OpenStack по HTTP
- OpenWeatherMap по HTTP
- Oracle Cloud по HTTP
- Palo Alto PA-440 по HTTP
- PHP-FPM по HTTP
- Podman по HTTP
- Proxmox VE по HTTP
- Pure Storage FlashArray v1 and v2 по HTTP
- RabbitMQ cluster по HTTP
- Ribbon SBC Edge по HTTP
- Ribbon SBC SWe CE по HTTP
- Ribbon SBC SWe Core по HTTP
- TiDB по HTTP
- TiDB PD по HTTP
- TiDB TiKV по HTTP
- Travis CI по HTTP
- Veeam Backup Enterprise Manager по HTTP
- Veeam Backup and Replication по HTTP
- VeloCloud SD-WAN по HTTP
- VeloCloud SD-WAN Edge по HTTP
- YugabyteDB по HTTP
- ZooKeeper по HTTP

Операции с шаблоном IPMI

Шаблоны IPMI не требуют какой-либо специальной настройки. Чтобы начать мониторинг, свяжите шаблон с целевым узлом сети (если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать шаблон).

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (открывается при нажатии на имя шаблона).

Доступный шаблон:

- Chassis by IPMI

Работа шаблона JMX

Шаги для обеспечения корректной работы шаблонов, собирающих метрики по JMX:

1. Убедитесь, что Zabbix Java gateway установлен и правильно настроен.
2. Привяжите шаблон к целевому узлу сети. На узле сети должен быть настроен интерфейс JMX. Если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать шаблон.
3. При необходимости скорректируйте значения макросов шаблона.
4. Настройте контролируемый экземпляр так, чтобы он разрешал обмен данными с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (открывается при нажатии на имя шаблона).

Доступны следующие шаблоны:

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX
- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX
- WildFly Domain by JMX
- WildFly Server by JMX

Работа шаблона ODBC

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики через мониторинг ODBC:

1. Убедитесь, что необходимый драйвер ODBC установлен на сервере Zabbix или прокси.
2. Привяжите шаблон к целевому узлу сети (если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон).
3. При необходимости настройте значения макросов шаблона.
4. Настройте контролируемый экземпляр так, чтобы он разрешал обмен данными с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (открывается при нажатии на имя шаблона).

Доступны следующие шаблоны:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Стандартизированные шаблоны для сетевых устройств

Обзор

Чтобы обеспечить мониторинг сетевых устройств, таких как коммутаторы и маршрутизаторы, мы создали две так называемые модели: для самого сетевого устройства (по сути, его шасси) и для сетевого интерфейса.

Для многих семейств сетевых устройств доступны шаблоны. Все шаблоны охватывают (где это возможно получить с устройства):

- Мониторинг неисправностей шасси (блоки питания, вентиляторы и температура, общий статус)
- Мониторинг производительности шасси (элементы данных CPU и памяти)
- Сбор инвентарной информации шасси (серийные номера, имя модели, версия прошивки)
- Мониторинг сетевого интерфейса с помощью IF-MIB и EtherLike-MIB (состояние интерфейса, нагрузка трафика интерфейса, состояние duplex для Ethernet)

Если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать шаблон.

Если вы импортируете новые шаблоны, поставляемые "из коробки", возможно, вам также потребуется обновить глобальное регулярное выражение `@Network interfaces for discovery` на:

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

чтобы исключить loopback- и null-интерфейсы на большинстве систем.

Устройства

Список семейств устройств, для которых доступны шаблоны:

Template name	Vendor	Device family	Known models	OS	MIBs used	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Сертифицировано
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Серия коммутаторов Aruba CX	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Сертифицировано
Brocade FC SNMP	Brocade	Коммутаторы Brocade FC	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Производительность Неисправность
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Сертифицировано
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Производительность Неисправность
Check Point Next Generation Firewall by SNMP	Check Point	Межсетевой экран следующего поколения		Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Сертифицировано

Template name	Vendor	Device family	Known models	OS	MIBs used	Tags
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Сертифицировано
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	Сертифицировано
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Сертифицировано
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T and 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Сертифицировано
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Сертифицировано
D-Link DES_DGS Switch SNMP	D-Link	Коммутаторы D-Link DES/DGX	DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Сертифицировано
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Производительность Неисправность Интерфейсы
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Сертифицировано
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Сертифицировано
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Производительность Инвентаризация
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Сертифицировано

Template name	Vendor	Device family	Known models	OS	MIBs used	Tags
HP Enterprise Switch SNMP	HP	Коммутатор HP Enterprise	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Сертифицировано
Huawei AR600 by SNMP	Huawei	Маршрутизаторы Huawei NetEngine серии AR600	Маршрутизаторы Huawei серий AR600/AR6100/AR6200/AR6300 с ПО VRP (V300R019/V300R021/V300R023 или новее)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Сертифицировано
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado с ПО V5		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Сертифицировано
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500/5600/6800/18500/18800 с ПО V6/V700		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Сертифицировано
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Сертифицировано
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Устройства Intel/QLogic Infiniband	Infiniband 12300		ICS-CHASSIS-MIB	Неисправность Инвентаризация
Juniper SNMP	Juniper	Модели MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Сертифицировано
Juniper MX NETCONF	Juniper	Модели MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Не применимо (использует NETCONF через SSH)*	Сертифицировано
Juniper MX SNMP	Juniper	Модели MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Сертифицировано

Template name	Vendor	Device family	Known models	OS	MIBs used	**Tags**
Mellanox SNMP	Mellanox	Устройства Mellanox Infini-band	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Сертифицировано
MikroTik CCR<device model> SNMP	MikroTik	Маршрутизаторы MikroTik Cloud Core Router (серия CCR)	Отдельные специализированные шаблоны доступны для MikroTik CCR1009-7G-1C-1S+, MikroTik CCR1009-7G-1C-1S+PC, MikroTik CCR1009-7G-1C-PC, MikroTik CCR1016-12G, MikroTik CCR1016-12S-1S+, MikroTik CCR1036-12G-4S-EM, MikroTik CCR1036-12G-4S, MikroTik CCR1036-8G-2S+, MikroTik CCR1036-8G-2S+EM, MikroTik CCR1072-1G-8S+, MikroTik CCR2004-16G-2S+, MikroTik CCR2004-1G-12S+2XS	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано

Template name	Vendor	Device family	Known models	OS	MIBs used	Tags
MikroTik CRS<device model> SNMP	MikroTik	Коммутаторы MikroTik Cloud Router Switch (серия CRS)	Отдельные специализированные шаблоны доступны для MikroTik CRS106-1C-5S, MikroTik CRS109-8G-1S-2HnD-IN, MikroTik CRS112-8G-4S-IN, MikroTik CRS112-8P-4S-IN, MikroTik CRS125-24G-1S-2HnD-IN, MikroTik CRS212-1G-10S-1S+IN, MikroTik CRS305-1G-4S+IN, MikroTik CRS309-1G-8S+IN, MikroTik CRS312-4C+8XG-RM, MikroTik CRS317-1G-16S+RM, MikroTik CRS326-24G-2S+IN, MikroTik CRS326-24G-2S+RM, MikroTik CRS326-24S+2Q+RM, MikroTik CRS328-24P-4S+RM, MikroTik CRS328-4C-20S-4S+RM, MikroTik CRS354-48G-4S+2Q+RM, MikroTik CRS354-48P-4S+2Q+RM	RouterOS/SwitchOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik CSS<device model> SNMP	MikroTik	Коммутаторы MikroTik Cloud Smart Switch (серия CSS)	Отдельные специализированные шаблоны доступны для MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Отдельные специализированные шаблоны доступны для MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано

Template name	Vendor	Device family	Known models	OS	MIBs used	Tags
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	Отдельные специализированные шаблоны доступны для MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	Отдельные специализированные шаблоны доступны для MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik RB<device model> SNMP	MikroTik	Маршрутизаторы серии MikroTik RB	Отдельные специализированные шаблоны доступны для MikroTik RB1100AHx4, MikroTik RB1100AHx4 Dude Edition, MikroTik RB2011iL-IN, MikroTik RB2011iL-RM, MikroTik RB2011iLS-IN, MikroTik RB2011UiAS-IN, MikroTik RB2011UiAS-RM, MikroTik RB260GS, MikroTik RB3011UiAS-RM, MikroTik RB4011iGS+RM, MikroTik RB5009UG+S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
MikroTik SNMP	MikroTik	Устройства MikroTik RouterOS	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицировано
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Неисправность Инвентаризация

Template name	Vendor	Device family	Known models	OS	MIBs used	Tags
QTech QSW SNMP	QTech	Устройства Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Производительность Инвентаризация
Stormshield SNS by SNMP	Stormshield	Межсетевые экраны Stormshield Network Security (SNS)	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Сертифицировано
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Производительность Инвентаризация
Ubiquiti AirOS SNMP	Ubiquiti	Беспроводные устройства Ubiquiti AirOS	NanoBridge,NanoStation,UniFi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Производительность
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Производительность Инвентаризация

Дизайн шаблонов

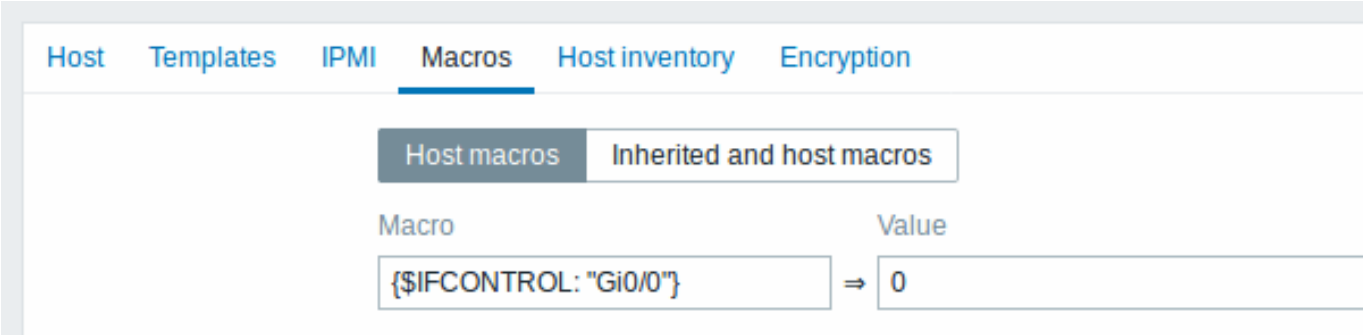
Шаблоны разработаны с учетом следующего:

- Пользовательские макросы используются где только возможно, поэтому триггеры можно подстроить;
- Низкоуровневые обнаружения используются где только возможно для минимизации количества неподдерживаемых элементов данных;
- Все шаблоны зависят от Template ICMP Ping, таким образом все устройства так же проверяются по протоколу ICMP;
- Элементы данных не используют MIB — в элементах данных и правилах обнаружения используются только SNMP OID. Поэтому для работы шаблонов нет необходимости загружать какие-либо MIB в Zabbix;
- Сетевые интерфейсы loopback при обнаружении фильтруются, также как и интерфейсы с ifAdminStatus = down(2)
- Используются 64-битные счётчики из IF-MIB::ifXTable, где это возможно. Если они не поддерживаются, по умолчанию вместо них используются 32-битные счётчики.

У всех обнаруженных сетевых интерфейсов имеется триггер, который следит за его рабочим состоянием (каналом связи), например:

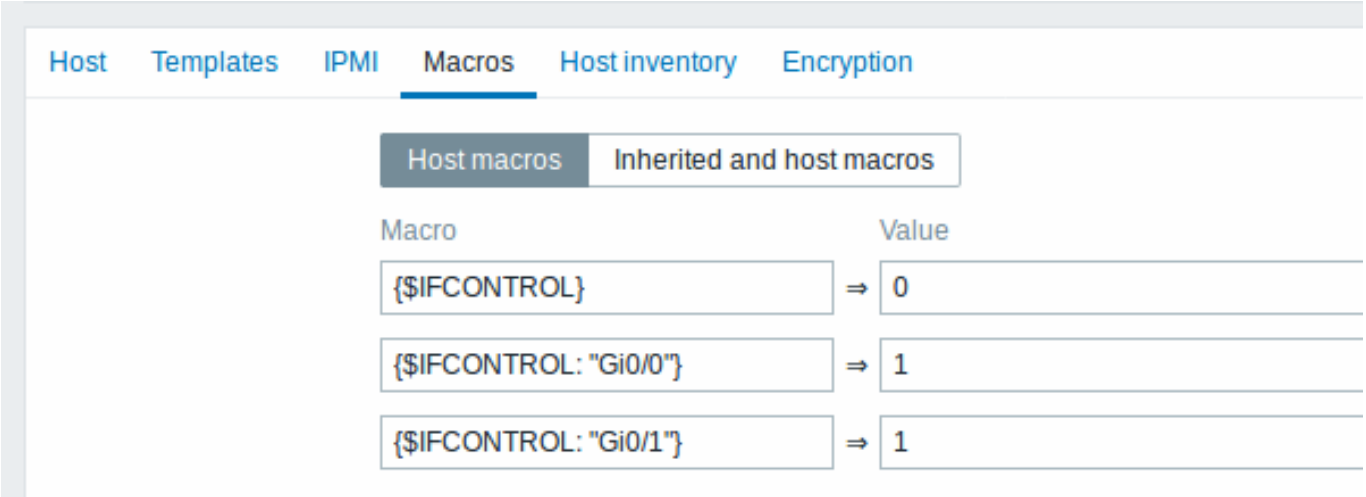
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])=
```

- Если вы не хотите мониторить это условие по какому-либо отдельному интерфейсу, создайте пользовательский макрос с контекстом со значением 0. Например:



где Gi0/0 является {#IFNAME}. Таким образом, триггер больше не будет использоваться для этого конкретного интерфейса.

- Вы также можете изменить поведение по умолчанию по всем триггерам (не срабатывать) и активировать этот триггер только по ограниченному количеству интерфейсов, например, только по магистральным каналам связи:



Теги

- Performance (Прозводительность) — MIB'ы семейства устройств дают возможность мониторинга элементов данных CPU и памяти;
- Fault (Ошибки) — MIB'ы семейства устройств дают возможность мониторинга по крайней мере одного датчика температуры;
- Inventory (Инвентарные данные) — MIB'ы семейства устройств дают возможность сбора по крайней мере серийного номера устройства и его наименование модели;
- Certified (Сертифицированный) — охватываются все три основные категории выше.

Операция шаблона VMware

Обзор

Zabbix включает ряд готовых к использованию шаблонов для мониторинга виртуальных машин, предназначенных для гипервизоров VMware vCenter и ESXi. Доступные шаблоны содержатся в двух отдельных наборах шаблонов:

- VMware — использует данные UUID для соответствующих макросов
- VMware FQDN — использует данные FQDN для соответствующих макросов

Шаблоны, предназначенные для мониторинга VMware vCenter или гипервизоров ESXi, содержат предварительно настроенные правила низкоуровневого обнаружения, а также различные встроенные проверки для мониторинга виртуальных установок.

Для корректной работы шаблона *VMware FQDN* каждая контролируемая виртуальная машина должна иметь уникальное имя ОС, соответствующее правилам FQDN. Кроме того, на каждой машине должны быть установлены VMware Tools/Open Virtual Machine Tools. Если эти предварительные условия выполнены, рекомендуется использовать шаблон *VMware FQDN*. Шаблон *VMware FQDN* доступен начиная с Zabbix 5.2, когда появилась возможность создавать узлы сети с пользовательскими интерфейсами.

 Также доступен классический шаблон *VMware*, который можно использовать, если требования FQDN не соблюдены. Однако у шаблона *VMware* есть известная проблема. Узлы сети для обнаруженных виртуальных машин создаются с именами, сохранёнными в vCenter (например, "VM1", "VM2" и т. д.). Если на этих узлах сети установлен агент Zabbix и включена активная авторегистрация агента Zabbix, процесс

авторегистрации будет считывать имена узлов сети в том виде, в котором они были зарегистрированы при запуске (например, ”vm1.example.com”, ”vm2.example.com” и т. д.). Это может привести к созданию новых узлов сети для уже существующих виртуальных машин (поскольку совпадений по именам не найдено), что приведёт к появлению дублирующихся узлов сети с разными именами.

Обратите внимание:

- Шаблон *VMware Hypervisor* можно как вручную привязать к узлу сети, так и использовать при обнаружении;
- Шаблон *VMware Guest* не следует вручную привязывать к узлу сети, его можно использовать только при обнаружении.

Конфигурация макросов узла сети

Чтобы использовать простые проверки VMware, для узла сети должны быть определены следующие пользовательские макросы:

- {$VMWARE.URL} - URL SDK службы VMware (vCenter или гипервизор ESXi) (https://servername/sdk)
- {$VMWARE.USERNAME} - имя пользователя службы VMware
- {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} службы VMware

Работа шаблонов OTLP

Zabbix предоставляет набор готовых шаблонов для сбора и мониторинга трассировок, метрик и журналов OpenTelemetry с помощью элементов данных типа *Telemetry query*.

Действия для обеспечения корректной работы шаблонов:

1. Проверьте файл `README` шаблона на наличие требований к конфигурации, специфичных для этого шаблона.
2. Создайте в Zabbix узел сети, соответствующий целевому объекту мониторинга.
3. Убедитесь, что параметр `TelemetryProvider` настроен в конфигурационном файле сервера Zabbix или прокси Zabbix.
4. Свяжите шаблон с созданным узлом сети (если шаблон недоступен в вашей установке Zabbix, сначала может потребоваться импортировать его).
5. При необходимости измените значения макросов на уровне шаблона.

Подробное описание шаблонов, включая полный список макросов, элементов данных и триггеров, доступно в файле `README` шаблона. Открыть этот файл можно, нажав на имя шаблона.

Доступны следующие шаблоны:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

Уведомления при событиях

Обзор

Предположим, мы уже настроили несколько элементов данных и триггеров и теперь мы получаем некоторые события, которые появляются как результат изменения состояния триггеров; настало время рассмотреть некоторые действия.

Начнем с того, что мы не хотим наблюдать за триггерами или за списком событий всё время. Было бы неплохо получать оповещения, если происходит что-то значительное (например, проблемы). Кроме того, когда возникают проблемы, мы бы хотели видеть, что все заинтересованные персоны проинформированы об этом.

Вот почему отправка оповещений является одной из ключевых задач, предлагаемой Zabbix. Можно указать, кто и когда должен быть оповещён при наступлении определённого события.

Чтобы была возможность отправлять и получать оповещения от Zabbix, вы должны:

- настроить какие-нибудь способы оповещения
- настроить действия, которые отправляют сообщения по одному из заданных способов оповещения

Действия состоят из *условий* и *операций*. В общем смысле, когда условия удовлетворяются, выполняются операции. Две основные операции — это отправка сообщений (оповещений) и выполнение удалённых команд.

Для событий, созданных обнаружениями и автоматической регистрацией, доступны несколько дополнительных операций. Они включают в себя добавление и удаление узлов сети, присоединение шаблонов и др.

Типы медиа

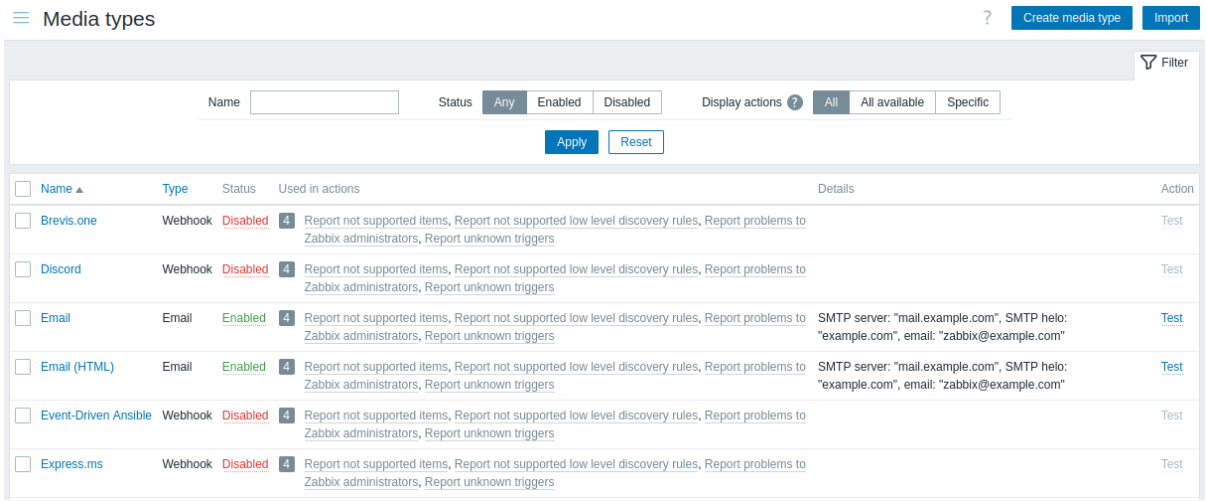
Обзор

Типы носителей — это каналы доставки, используемые для отправки уведомлений и оповещений из Zabbix.

Типы носителей поддерживают следующие способы доставки:

- Электронная почта
- Push-уведомление
- SMS
- Пользовательский скрипт
- Вебхук

Типы носителей настраиваются в разделе *Оповещения* > *Типы носителей*. Некоторые типы носителей предварительно определены в наборе данных по умолчанию. Вам нужно только настроить их параметры, чтобы они заработали.



Чтобы понять, какое место типы носителей занимают в процессе оповещения, рассмотрим три требования для доставки уведомлений из Zabbix конечным пользователям:

1. Должна быть определена операция действия, отправляющая уведомления.
2. Должен быть определен рабочий тип носителя (например, *Электронная почта*, отправляющая оповещения через SMTP).
3. В носителях пользователя должны быть указаны данные доставки на уровне пользователя (например, адреса электронной почты, номера телефонов и т. д.).

Тестирование типа средства оповещения

Чтобы проверить работу настроенного типа средства оповещения, нажмите *Тест* в списке типов средств оповещения.

Запрос на тестирование будет отправлен на сервер Zabbix. Сервер Zabbix попытается отправить оповещение с использованием указанного типа средства оповещения и вернет результат в веб-интерфейс. Веб-интерфейс будет ожидать, пока сервер вернет результаты. По умолчанию время ожидания тестирования типа средства оповещения составляет 65 секунд (настраивается в разделе *Администрирование* > *Общие* > *Тайм-ауты*).

См. также сведения о тестировании:

- Электронная почта
- Push-уведомление
- Вебхук
- Скрипт

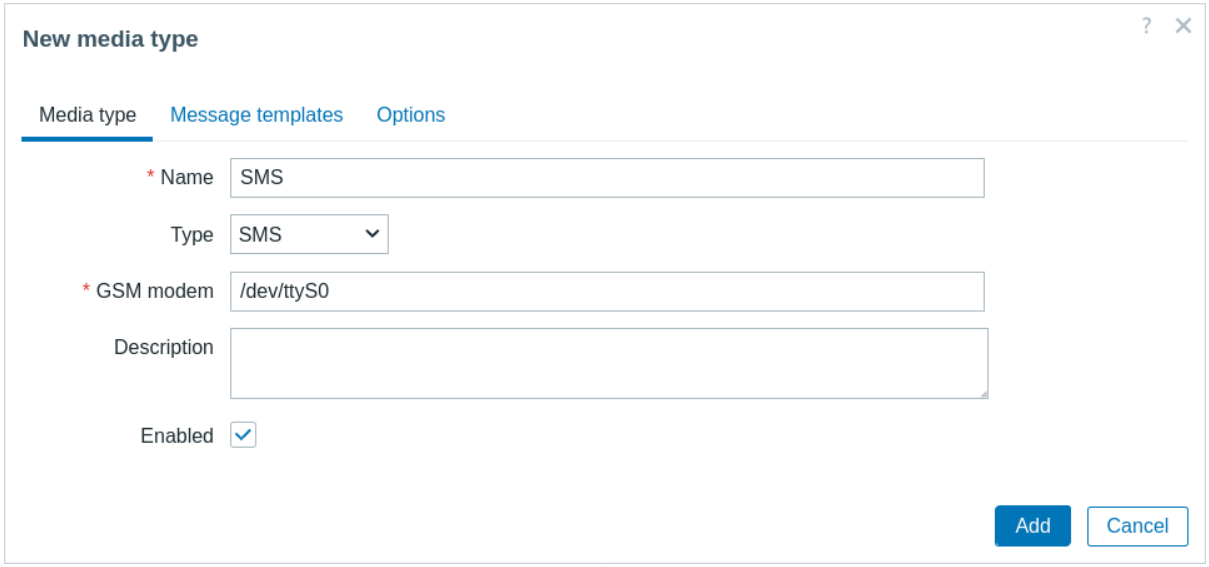
Настройка

Чтобы создать тип носителя в веб-интерфейсе Zabbix:

1. Перейдите в раздел *Оповещения* > *Типы носителей*.
2. Нажмите *Создать тип носителя*.
3. Введите параметры типа носителя в форме.

Некоторые параметры являются **общими** для всех способов доставки.

Note:
Если один или несколько способов доставки указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса (`zabbix.conf.php`), эти способы доставки недоступны в списке *Тип*.

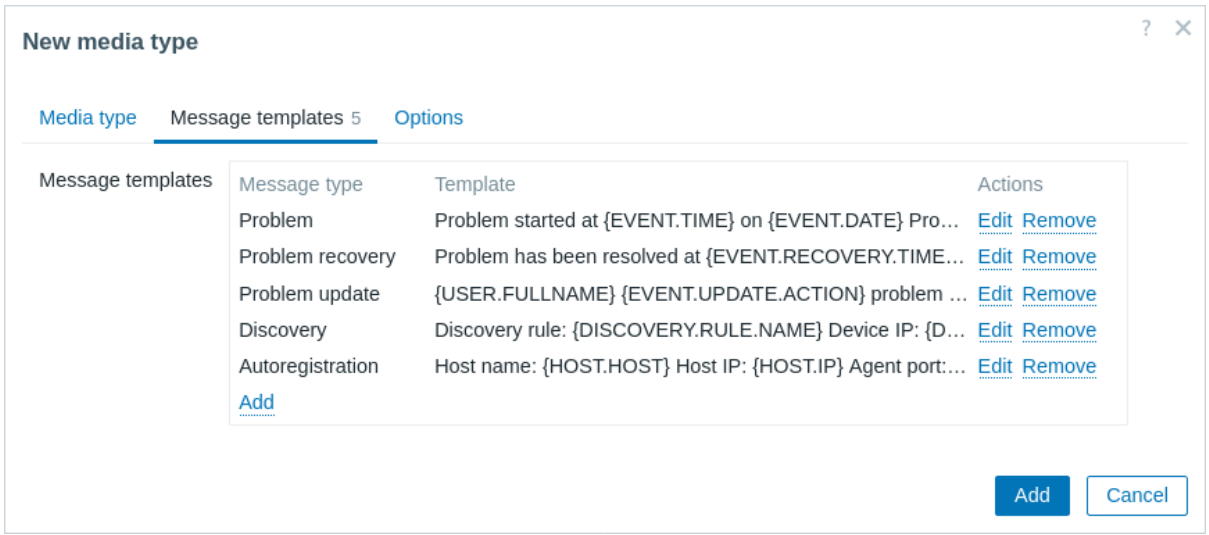


Параметр	Описание
Название	Название типа носителя.
Тип	Выберите способ доставки для типа носителя.
Описание	Введите описание типа носителя.
Включен	Установите флажок, чтобы включить тип носителя.

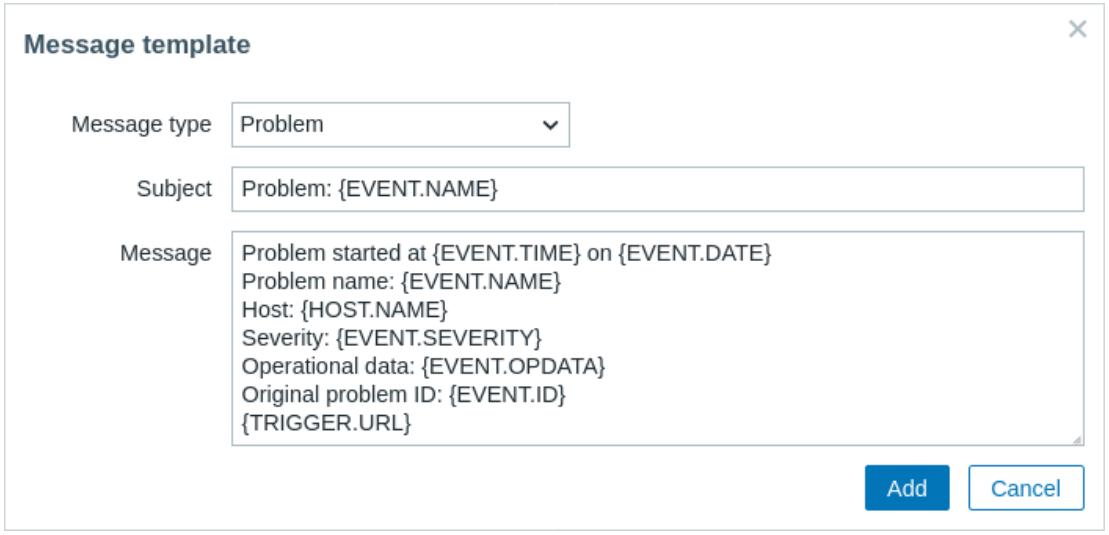
Параметры, специфичные для отдельных способов доставки, см. на страницах электронная почта, push-уведомление, SMS, пользовательский alertscript или вебхук.

Шаблоны сообщений

Вкладка **Шаблоны сообщений** содержит сообщения по умолчанию для каждого типа события (проблема, восстановление после проблемы, обнаружение и т. д.).



Нажмите **Добавить**, чтобы задать сообщение по умолчанию (или **Изменить**, чтобы обновить существующее сообщение):



Parameter	Description
Message type	Тип события, для которого должно использоваться сообщение по умолчанию. Для каждого типа события можно определить только одно сообщение по умолчанию.
Subject	Тема сообщения по умолчанию. Тема может содержать макросы. Ее длина ограничена 255 символами. Тема недоступна для типа носителя SMS.
Message	Сообщение по умолчанию. Его длина ограничена определенным количеством символов в зависимости от типа базы данных (см. Отправка сообщений для получения дополнительной информации). Сообщение может содержать поддерживаемые макросы. В сообщениях о проблеме и обновлении проблемы поддерживаются макросы выражений (например, `{?avg(/host/key,1h)}`).

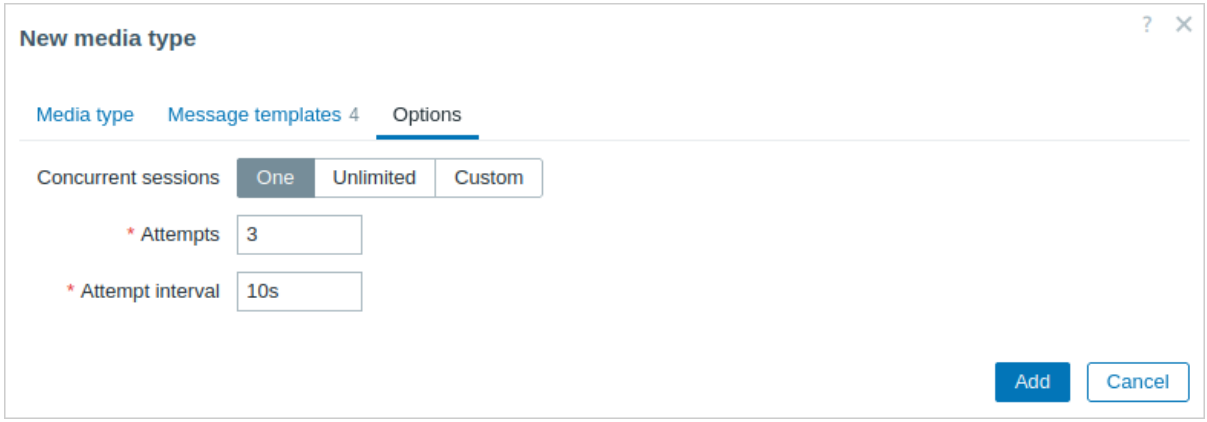
Обратите внимание, что сообщения по умолчанию переопределяются пользовательскими сообщениями, если они заданы в операциях действия.

Warning:
Определение шаблонов сообщений обязательно для всех способов доставки, включая вебхуки или пользовательские скрипты оповещений, которые не используют сообщения по умолчанию для уведомлений. Например, действие "Send message to Pushover webhook" не сможет отправить уведомления о проблеме, если сообщение о проблеме для вебхука Pushover не определено.

Параметры

Вкладка **Параметры** содержит настройки обработки оповещений.
Для каждого типа носителя можно настроить один и тот же набор параметров.

Все типы носителей обрабатываются параллельно.
Максимальное количество одновременных сеансов можно настроить для каждого типа носителя,
однако общее количество процессов отправки оповещений на сервере можно ограничить только с помощью параметра StartAlerters.
Оповещения, созданные одним триггером, обрабатываются последовательно.
Таким образом, несколько уведомлений могут обрабатываться одновременно только в том случае, если они созданы несколькими триггерами.



Параметр	Описание
Одновременные сеансы	Выберите количество параллельных сеансов отправки оповещений для типа носителя: **Один** — один сеанс **Неограниченно** — неограниченное количество сеансов **Пользовательское** — выберите пользовательское количество сеансов Неограниченные или высокие значения означают большее количество параллельных сеансов и увеличение пропускной способности при отправке уведомлений. Неограниченные или высокие значения следует использовать в крупных средах, где может потребоваться одновременная отправка большого количества уведомлений. Если уведомлений требуется отправить больше, чем доступно одновременных сеансов, оставшиеся уведомления будут помещены в очередь; они не будут потеряны.
Попытки	Количество попыток отправки уведомления. Можно указать до 100 попыток; значение по умолчанию — «3». Если указано «1», Zabbix отправит уведомление только один раз и не будет повторять попытку в случае ошибки отправки.
Интервал между попытками	Частота повторных попыток отправки уведомления в случае ошибки отправки, в секундах (0–3600). Если указано «0», Zabbix немедленно повторит попытку. Поддерживаются суффиксы времени (например, 30s, 1m, 1h).

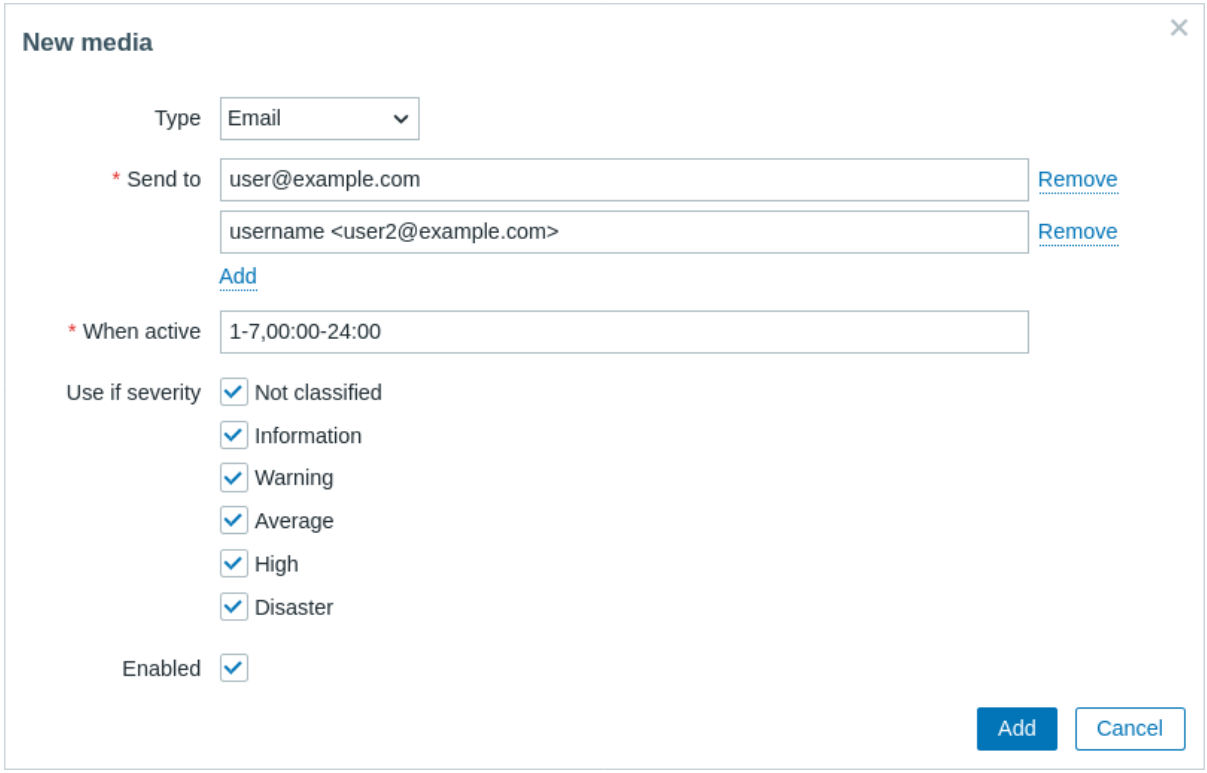
Пользовательские средства доставки

Если типы средств доставки определяют **как** будет отправлено уведомление, то пользовательские средства доставки определяют **куда** необходимо отправить уведомление.

Пользовательские средства доставки (например, адрес электронной почты, идентификатор пользователя вебхука и т. д.) должны быть указаны в профиле пользователя независимо от способа доставки. Действие, отправляющее сообщения пользователю *Admin* с помощью вебхука *X*, не сможет доставить их, если сведения о доставке через вебхук *X* не указаны в профиле пользователя Admin.

Чтобы определить пользовательские средства доставки:

1. Перейдите в раздел *Пользователи* > *Пользователи* и откройте форму свойств пользователя. Также можно перейти в раздел *Настройки пользователя* > *Уведомления* собственного профиля пользователя.
2. Нажмите *Добавить* на вкладке *Средства доставки*.
3. Укажите параметры пользовательского средства доставки в форме.



Параметр	Описание
Тип	В раскрывающемся списке отображаются названия включенных типов средств доставки. Обратите внимание: при редактировании средства доставки отключенного типа его название отображается красным цветом.
Кому	Укажите контактные данные, на которые следует отправлять сообщения. Для типа средства доставки «Электронная почта» можно добавить несколько адресов, нажав кнопку Add под полем адреса. В этом случае уведомления будут отправляться на все указанные адреса. Примеры адресов приведены в описании параметра *Электронная почта* для типа средства доставки email. Для типа средства доставки «Push-уведомление» можно выбрать активные мобильные устройства пользователя или добавить одно или несколько определенных устройств.
Когда активно	Можно ограничить время отправки сообщений, например указать только рабочие дни (1-5,09:00-18:00). Обратите внимание: это ограничение основано на часовом поясе пользователя. Если часовой пояс пользователя изменен и отличается от часового пояса системы, может потребоваться соответствующим образом скорректировать это ограничение, чтобы не пропустить важные сообщения. Описание формата приведено на странице Задание периода времени. Поддерживаются пользовательские макросы.
Использовать при важности	Установите флажки уровней важности триггеров, для которых необходимо получать уведомления. *Примечание*: уровень важности по умолчанию («Не классифицировано») **должен быть** выбран, если необходимо получать уведомления о событиях, не связанных с триггерами. После сохранения выбранные уровни важности триггеров будут отображаться соответствующими цветами важности, а невыбранные будут выделены серым цветом.
Состояние	Состояние пользовательского средства доставки. **Включено** - используется. **Отключено** - не используется.

Электронная почта

Обзор

Чтобы настроить email в качестве канала доставки сообщений, вам необходимо настроить email как способ оповещений и назначить соответствующие адреса пользователям.

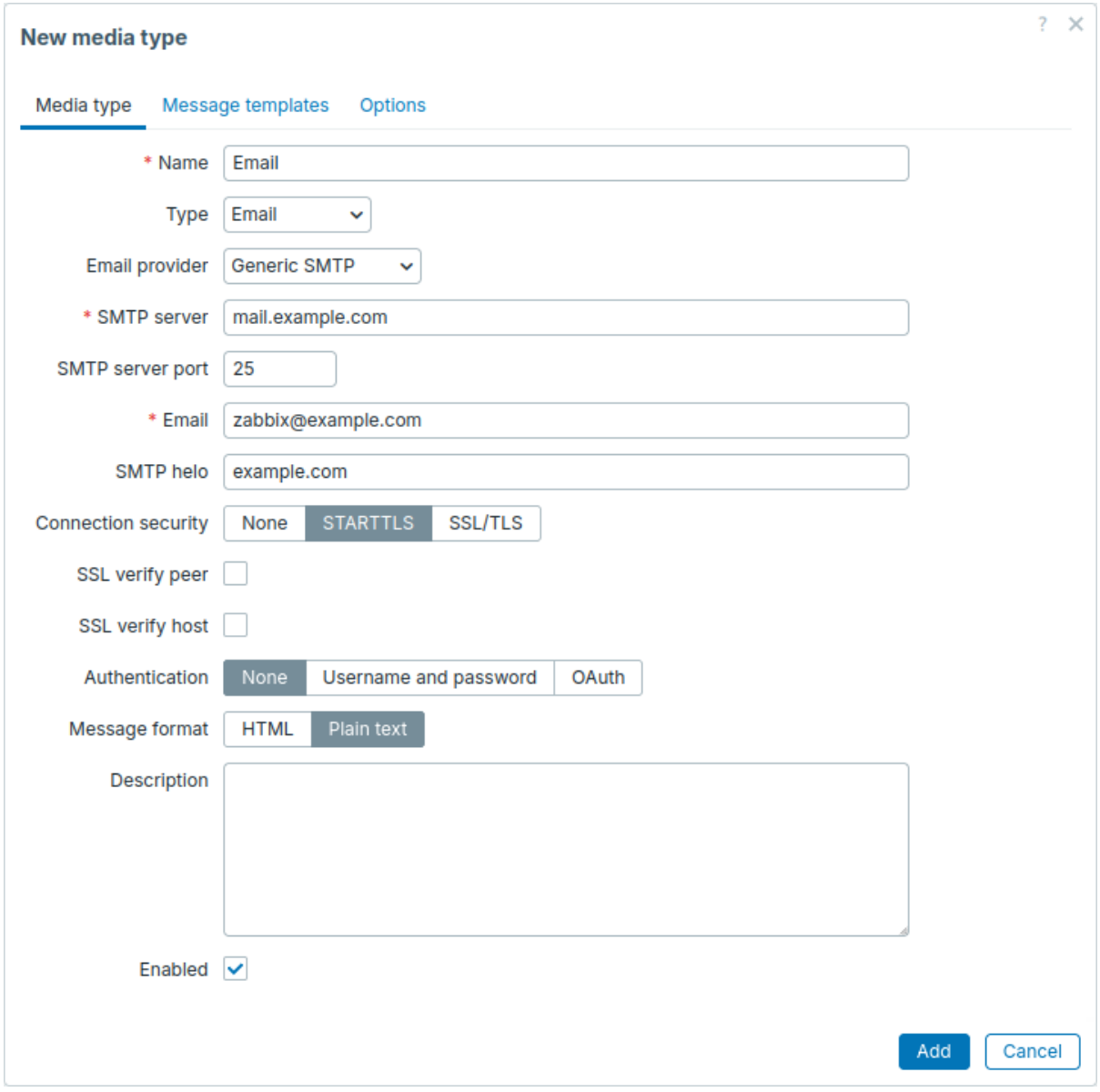
Note:
Несколько оповещений по одному событию группируются вместе в одной ветке email сообщений.

Настройка

Чтобы настроить электронную почту в качестве типа носителя:

1. Перейдите в раздел *Оповещения* > *Типы носителей*.
2. Нажмите *Создать тип носителя* (или выберите *Email* в списке предварительно определенных типов носителей).
3. Введите параметры типа носителя электронной почты в форме.

Вкладка **Тип носителя** содержит общие атрибуты типа носителя:



Все обязательные поля отмечены красной звездочкой.

Warning:
При экспорте конфигурации типа носителя электронной почты пароль экспортируется в открытом виде.

Следующие параметры относятся к типу носителя электронной почты:

Параметр	Описание
Поставщик электронной почты	Выберите поставщика электронной почты: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* или *Office365 relay*. Если выбрать вариант, связанный с Gmail или Office365, потребуется указать только адрес электронной почты отправителя и пароль; такие параметры, как *SMTP-сервер*, *Порт SMTP-сервера*, *SMTP helo* и *Безопасность соединения*, будут автоматически заполнены Zabbix. См. также: Автоматизированные типы носителей Gmail/Office365.
SMTP-сервер	Укажите SMTP-сервер для обработки исходящих сообщений. Это поле доступно, если в качестве поставщика электронной почты выбран *Generic SMTP*.
Порт SMTP-сервера	Укажите порт SMTP-сервера для обработки исходящих сообщений. Это поле доступно, если в качестве поставщика электронной почты выбран *Generic SMTP*.
Email	Указанный здесь адрес будет использоваться в качестве адреса **From** для отправляемых сообщений. Поддерживается добавление отображаемого имени отправителя к фактическому адресу электронной почты (например, `Zabbix_info <zabbix@company.com>`). В Zabbix существуют некоторые ограничения на отображаемые имена в сообщениях электронной почты по сравнению с ограничениями, разрешенными RFC 5322, как показано в следующих примерах. Допустимые примеры: `zabbix@company.com` (только адрес электронной почты; угловые скобки не требуются) `Zabbix_info <zabbix@company.com>` (отображаемое имя, за которым следует адрес электронной почты в угловых скобках) `Ω-monitoring <zabbix@company.com>` (символы UTF-8 в отображаемом имени) Недопустимые примеры: `Zabbix HQ zabbix@company.com` (отображаемое имя указано, но адрес электронной почты не заключен в угловые скобки) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (хотя это допустимо согласно RFC 5322, цитируемые пары и комментарии не поддерживаются в сообщениях электронной почты Zabbix).
SMTP helo	Укажите корректное значение SMTP helo, обычно имя домена. Если поле пусто, будет отправлено доменное имя электронной почты (то есть значение после `@` в поле *Email*). Если получить доменное имя невозможно, в журнал будет записано предупреждение уровня отладки, а в качестве домена для команды HELO будет отправлено имя узла сети сервера. Это поле доступно, если в качестве поставщика электронной почты выбран *Generic SMTP*.
Безопасность соединения	Выберите уровень безопасности соединения: **None** - не использовать параметр CURLOPT_USE_SSL **STARTTLS** - использовать параметр CURLOPT_USE_SSL со значением CURLUSESSL_ALL **SSL/TLS** - использование CURLOPT_USE_SSL необязательно.
Проверять SSL-сертификат узла	Установите флажок, чтобы проверять SSL-сертификат SMTP-сервера. Значение директивы конфигурации сервера "SSLCALocation" следует указать в CURLOPT_CAPATH для проверки сертификата. Это устанавливает параметр cURL CURLOPT_SSL_VERIFYPEER.
Проверять имя узла SSL	Установите флажок, чтобы проверить соответствие поля *Common Name* или *Subject Alternate Name* сертификата SMTP-сервера. Это устанавливает параметр cURL CURLOPT_SSL_VERIFYHOST.
Аутентификация	Выберите уровень аутентификации: **None** - параметры cURL не устанавливаются **Username and password** - подразумевает "AUTH=*", оставляя выбор механизма аутентификации cURL **OAuth** - аутентификация OAuth Аутентификация OAuth не поддерживается для поставщика электронной почты *Office365 relay*.
Имя пользователя	Имя пользователя для аутентификации. Устанавливает значение CURLOPT_USERNAME. Поддерживаются пользовательские макросы.

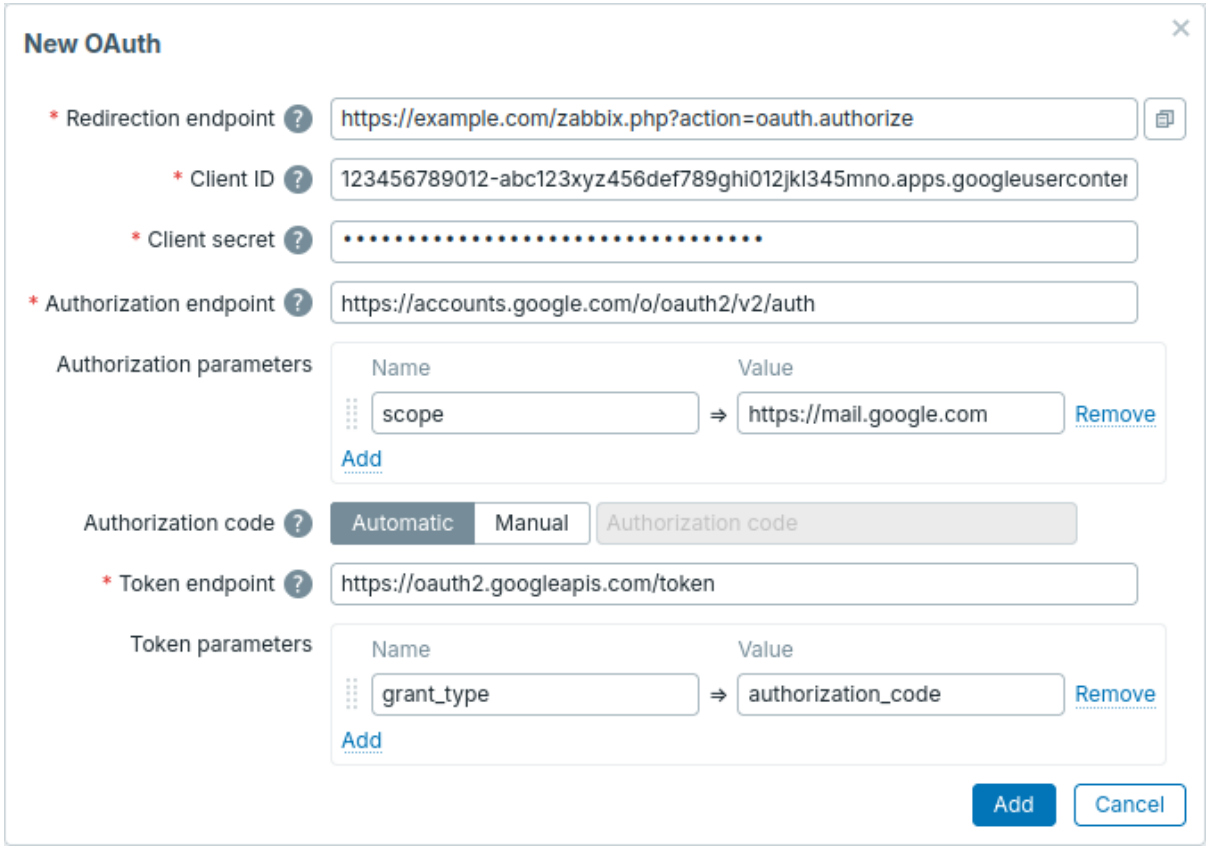
Параметр	Описание
Пароль	Пароль для аутентификации. Устанавливает значение CURLOPT_PASSWORD. Поддерживаются пользовательские макросы. Указанное здесь значение экспортируется без изменений при экспорте конфигурации типа носителя. Чтобы не раскрывать конфиденциальные учетные данные в экспортируемых файлах, используйте пользовательский макрос (например, `{$EMAIL_SMTP_PASSWORD}`) вместо буквального пароля — будет экспортирована ссылка на макрос, а секретное значение следует настроить заново в целевой системе.
Токены OAuth	Нажмите *Настроить*, чтобы задать параметры получения токенов OAuth в новом окне. Это поле доступно только если в поле *Аутентификация* выбран вариант "OAuth".
Формат сообщения	Выберите формат сообщения: **HTML** - отправлять в формате HTML; **Plain text** - отправлять как обычный текст.

Attention:
Для включения аутентификации SMTP сервер Zabbix должен быть скомпилирован с параметром `--with-libcurl` компиляции, который требует libcurl (подробные сведения о версиях см. в требованиях к серверу).

Подробнее о настройке сообщений по умолчанию и параметров обработки оповещений см. в разделе общие параметры типов носителей.

OAuth tokens

Для получения OAuth tokens требуются следующие параметры:



Warning:
Client secret экспортируется в открытом виде при экспорте конфигурации типа медиа email.

Parameter	Description
Redirection endpoint	Введите URL веб-интерфейса Zabbix, на который служба OAuth будет перенаправлять ответ после авторизации OAuth (в формате `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Он автоматически задается для нового типа медиа email, если определен URL веб-интерфейса.
Client ID	Введите уникальный идентификатор клиентского приложения, зарегистрированного на сервере авторизации OAuth.
Client secret	Введите закрытый секрет клиентского приложения, зарегистрированного на сервере авторизации OAuth.
Authorization endpoint	Введите URL сервера авторизации OAuth для запроса авторизации пользователя.
Authorization parameters	Введите параметры для конечной точки авторизации.
Authorization code	Введите код авторизации: **Automatic** - код будет получен автоматически через страницу перенаправления; **Manual** - введите код вручную, если автоматическое получение не удалось.
Token endpoint	Введите URL сервера авторизации OAuth для обмена кода авторизации на токены доступа и обновления.
Token parameters	Введите параметры для access token.

Note:
Получение OAuth tokens имеет автоматизированные возможности для почтовых провайдеров **Gmail**, **Gmail relay** и **Office365**. В этом случае требуется указать только значения параметров *Redirection endpoint*, *Client ID* и *Client secret*. Zabbix автоматически заполнит остальные необходимые значения (см. значения OAuth URL по умолчанию для провайдера).

Получение токенов доступа и обновления OAuth

После отправки параметров OAuth:

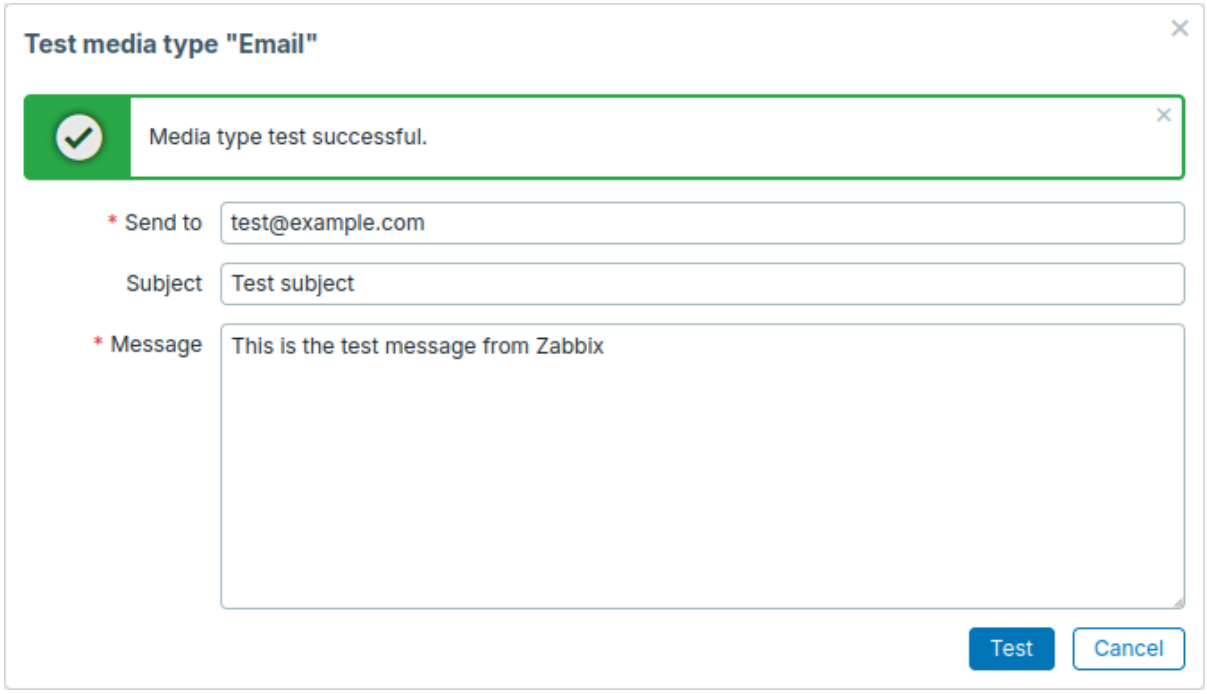
1. Открывается всплывающее окно браузера, перенаправляющее пользователя к *Authorization endpoint*.
2. Пользователь авторизует Zabbix в сервисе OAuth.
3. Сервис OAuth перенаправляет пользователя к действию веб-интерфейса Zabbix `oauth.authorize` с кодом авторизации и значением области действия.
4. В ответ Zabbix выполнит запрос к *Token endpoint*, чтобы обменять код авторизации на токены доступа и обновления.

Тестирование

Чтобы проверить, что настроенный тип медиа для электронной почты работает корректно:

1. Найдите нужный тип медиа для электронной почты в списке типов медиа.
2. Нажмите *Test* в последнем столбце списка (откроется окно тестирования).
3. Укажите адрес получателя в поле *Send to*, текст сообщения и, при необходимости, тему.
4. Нажмите *Test*, чтобы отправить тестовое сообщение.

В том же окне будет отображено сообщение об успешном выполнении или ошибке теста:



Пользовательские способы оповещения

После настройки типа оповещений *Email* перейдите в раздел *Пользователи* > *Пользователи* и отредактируйте профиль пользователя, чтобы назначить пользователю способ оповещения по электронной почте. Шаги по настройке пользовательских способов оповещения, общие для всех типов способов оповещения, описаны на странице Типы способов оповещения.

Автоматизированные типы медиа Gmail/Office365

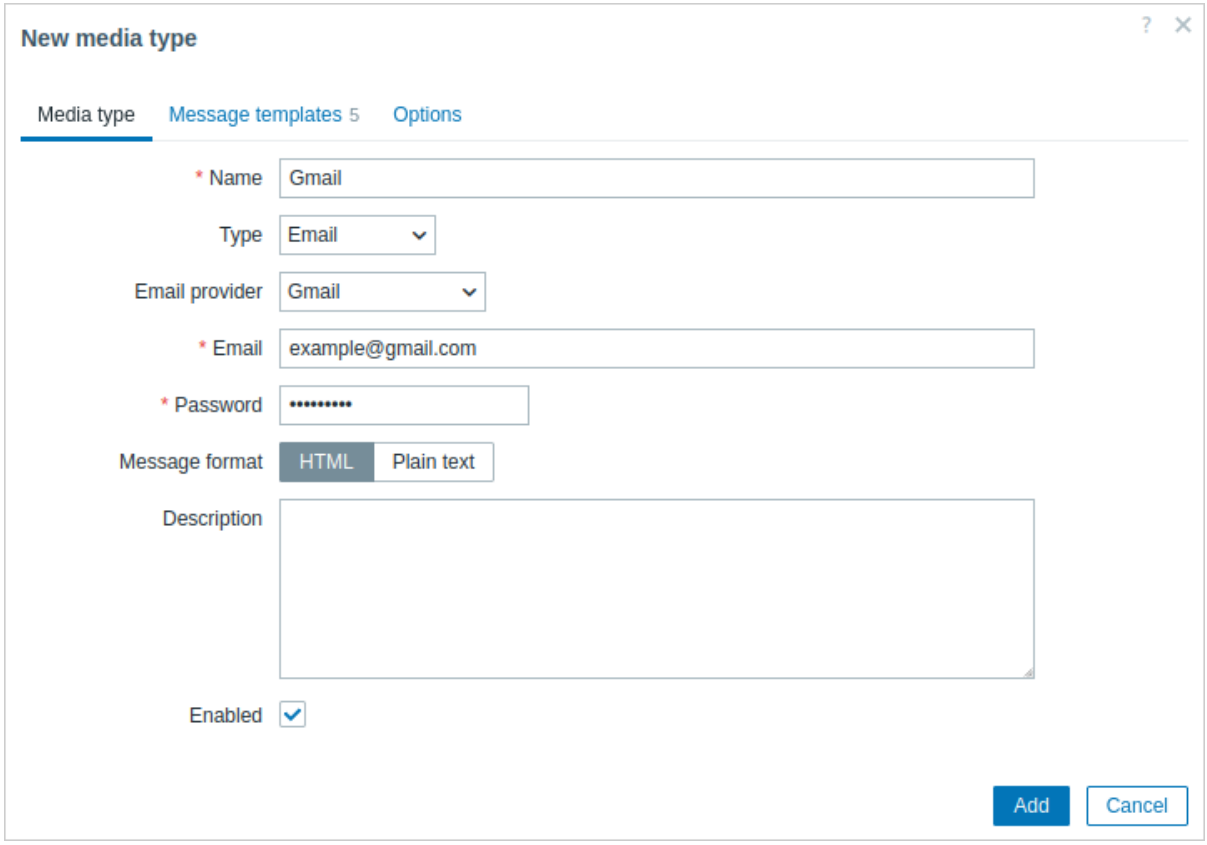
Обзор

Пользователи Gmail или Office365 могут воспользоваться автоматизированными возможностями в настройке типа оповещения.

Настройка

Поле *Поставщик электронной почты* в настройках типа носителя электронной почты позволяет выбрать предварительно настроенные параметры для Gmail и Office 365.

При выборе параметров, связанных с Gmail/Office365, для создания рабочего типа носителя достаточно указать адрес электронной почты и пароль отправителя.



После указания адреса электронной почты и пароля Zabbix сможет автоматически заполнить все необходимые настройки типов носителей Gmail/Office365 фактическими или рекомендуемыми значениями: *SMTP-сервер*, *порт SMTP-сервера*, *SMTP HELO* и *Безопасность соединения*. Благодаря этой автоматизации данные поля даже не отображаются, однако SMTP-сервер и сведения об электронной почте можно просмотреть в списке типов носителей (см. столбец *Подробности*).

Также обратите внимание:

- Для параметров ретрансляции пароль не требуется.
- Для ретрансляции Office365 доменное имя указанного адреса электронной почты будет использоваться для динамического заполнения SMTP-сервера (то есть значение "`example.com`" в `example-com.mail.protection.outlook.com` будет заменено фактическим значением).

Токены OAuth

Получение токенов OAuth имеет автоматизированные возможности для почтовых провайдеров *Gmail*, *Gmail relay* и *Office365*.



Для получения доступа OAuth и токена обновления необходимо указать только ограниченный набор параметров — *Redirection endpoint*, *Client ID* и *Client secret*. Zabbix автоматически заполнит остальные необходимые значения (см. URL OAuth по умолчанию в зависимости от провайдера).

Обратите внимание, что для настройки авторизации OAuth для этих провайдеров также можно использовать форму Generic SMTP.

Note:
SmtpClientAuthentication должен быть включен либо для каждого пользователя/общего почтового ящика, либо для всего арендатора в Office365 (по умолчанию не включен).

Значения OAuth URL по умолчанию в зависимости от провайдера

В следующей таблице перечислены значения OAuth URL и параметры по умолчанию для каждого провайдера, используемые Zabbix.

Параметр	Gmail	Office365	Generic SMTP
Конечная точка авторизации	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authorize`	Нет значения по умолчанию
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	Нет значения по умолчанию
access_type	`offline`	Не используется	Нет значения по умолчанию
prompt	`consent`	Не используется	Нет значения по умолчанию
redirect_uri	Значение поля *Redirection endpoint* автоматически добавляется кодом. Не указано в параметрах *Authorization endpoint*.		
state	Уникальный хеш для идентификации способа оповещения, обновляемого с помощью OAuth-токена, автоматически добавляется кодом. Не указан в параметрах *Authorization endpoint*.		
Конечная точка токена	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Нет значения по умолчанию
grant_type	`authorization_code`	`authorization_code`	Нет значения по умолчанию
redirect_uri	Значение поля *Redirection endpoint* автоматически добавляется кодом. Не указано в параметрах *Token endpoint*.		
state	Уникальный хеш для идентификации способа оповещения, обновляемого с помощью OAuth-токена, автоматически добавляется кодом. Не указан в параметрах *Token endpoint*.		

Обходное решение Office365 для личных учетных записей

Личные учетные записи Office365 не поддерживают строки запроса в URL перенаправления.

Чтобы обойти эту проблему, администратор веб-сервера Apache может добавить следующее правило rewrite в файл `.htaccess`:

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Это позволяет использовать упрощенный URL перенаправления, например:

```
http://server-name/zabbix/oauth.authorize
```

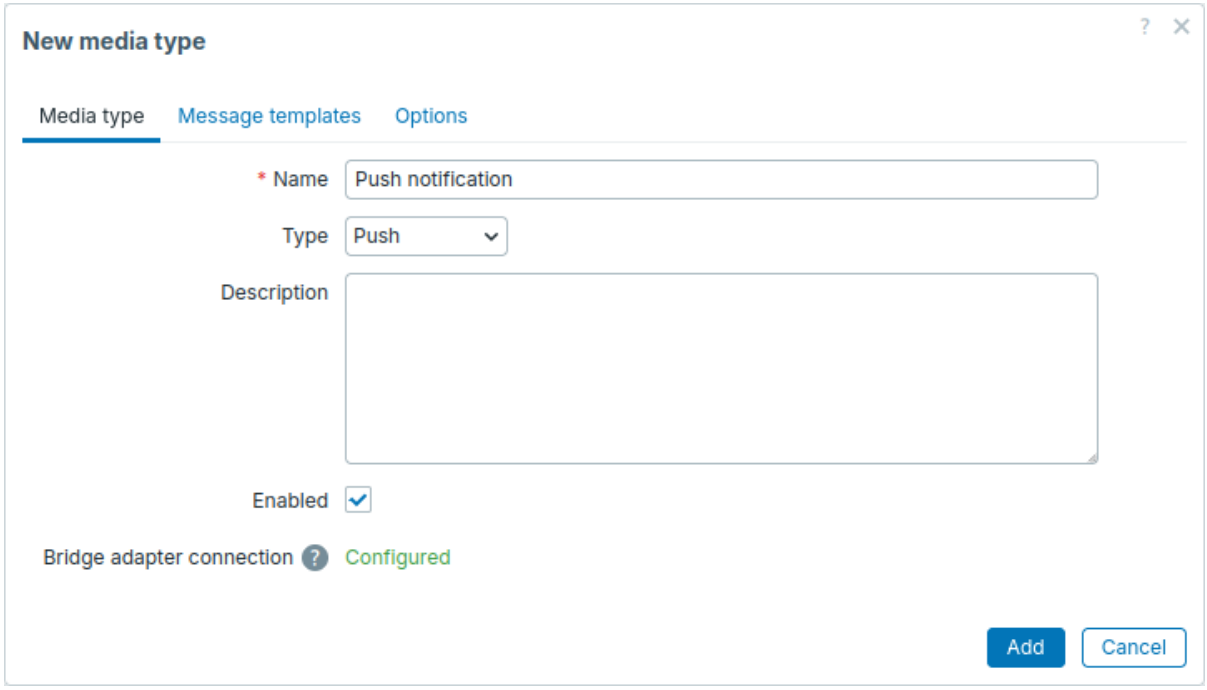
вместо URL с параметрами запроса, обеспечивая совместимость с личными учетными записями Office365.

Push-уведомление

Этот тип носителя позволяет серверу Zabbix отправлять push-уведомления в мобильное приложение Zabbix на вашем мобильном устройстве.

Настройка

При создании типа носителя выберите *Push* в поле *Type*:



Форма настройки содержит общие параметры, шаблоны сообщений и опции типа носителя.

В форме также отображается, настроен ли адаптер моста Zabbix.
 Если адаптер не настроен, вы всё равно можете создать тип носителя, но он не будет работать.

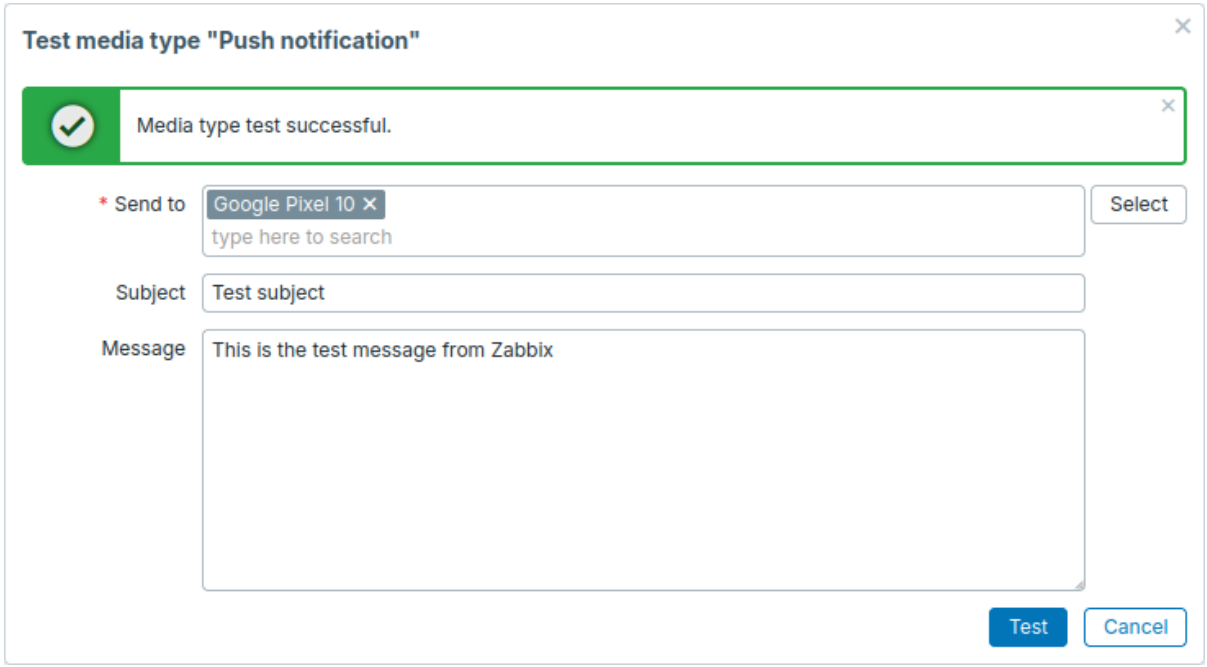
После создания и включения типа носителя добавьте устройство пользователю, чтобы автоматически добавить тип носителя этому пользователю (или добавьте носитель пользователя вручную). Это позволит пользователю получать push-уведомления на мобильном устройстве.

Тестирование

Чтобы проверить работу этого типа носителя:

1. Найдите тип носителя в списке типов носителей.
2. Нажмите *Тест* в последнем столбце списка, чтобы открыть форму тестирования.
3. В поле *Отправить на* выберите устройство.
4. Нажмите *Тест*, чтобы отправить тестовое push-уведомление на выбранное устройство.

В этом же окне появится сообщение об успешной или неуспешной отправке:



SMS

Обзор

Zabbix поддерживает отправку SMS сообщений с использованием GSM модемов, подключённых к последовательному порту Zabbix сервера.

Убедитесь, что:

- Скорость последовательного устройства (обычно /dev/ttyS0 под Linux) совпадает со скоростью GSM-модема. Zabbix не устанавливает скорость для последовательного соединения. Он использует настройки по умолчанию.
- Пользователь «zabbix» имеет доступ на чтение/запись к последовательному устройству. Выполните команду ls -l /dev/ttyS0 для просмотра текущих прав доступа к последовательному устройству.
- На GSM модеме введён ПИН-код, и он сохраняется даже после перезагрузки. В качестве альтернативного варианта, можно отключить запрос ПИН-кода на самой SIM карте. ПИН-код можно ввести путём использования команды AT+CPIN="NNNN" (где NNNN — это ваш ПИН-код, кавычки должны присутствовать) в терминальном приложении, таком как Unix minicom или Windows HyperTerminal.

Zabbix был протестирован на следующих GSM модемах:

- Siemens MC35
- Teltonika ModemCOM/G10

Для настройки SMS как канала отправки сообщений, вам также необходимо настроить SMS в качестве способа оповещений и ввести соответствующие номера телефонов для пользователей.

Настройка

Чтобы настроить SMS в качестве типа средства оповещения:

1. Перейдите в раздел *Оповещения* > *Типы средств оповещения*.
2. Нажмите *Создать тип средства оповещения* (или выберите *SMS* в списке предварительно определённых типов средств оповещения).
3. Укажите параметры типа средства оповещения SMS в форме.

Следующие параметры относятся к типу средства оповещения SMS:

Параметр	Описание
GSM-модем	Укажите имя последовательного устройства GSM-модема. Введённый здесь путь будет проверен на соответствие параметру сервера SMSDevices (если он указан).

Подробнее о настройке сообщений по умолчанию и параметров обработки оповещений см. в разделе общие параметры типов средств оповещения. Обратите внимание, что параллельная обработка отправки SMS-уведомлений невозможна.

Пользовательские способы оповещения

После настройки типа носителя SMS перейдите в раздел *Пользователи* > *Пользователи* и отредактируйте профиль пользователя, чтобы назначить ему SMS-носитель.

Шаги по настройке пользовательских способов оповещения, общие для всех типов носителей, описаны на странице Типы носителей.

Пользовательские скрипты оповещений

Обзор

Если вас не устраивают существующие типы медиа для отправки оповещений, есть альтернативный способ сделать это. Вы можете создать скрипт, который будет обрабатывать уведомление нужным вам образом.

Пользовательские скрипты оповещений выполняются на сервере Zabbix. Эти скрипты должны находиться в каталоге, указанном в параметре AlertScriptsPath файла конфигурации сервера.

Ниже приведен пример пользовательского скрипта оповещения:

```
#####!/bin/bash

to=$1
subject=$2
```

```
body=$3
host=$4
value=$5

cat <<EOF | mail -s "$subject" "$to"
$body

Host: $host
Value: $value
EOF
```

Attention:
Zabbix проверяет код завершения выполненных команд и скриптов. Любой код завершения, отличный от **0**, считается ошибкой выполнения команды. В таких случаях Zabbix попытается повторить неудачное выполнение.

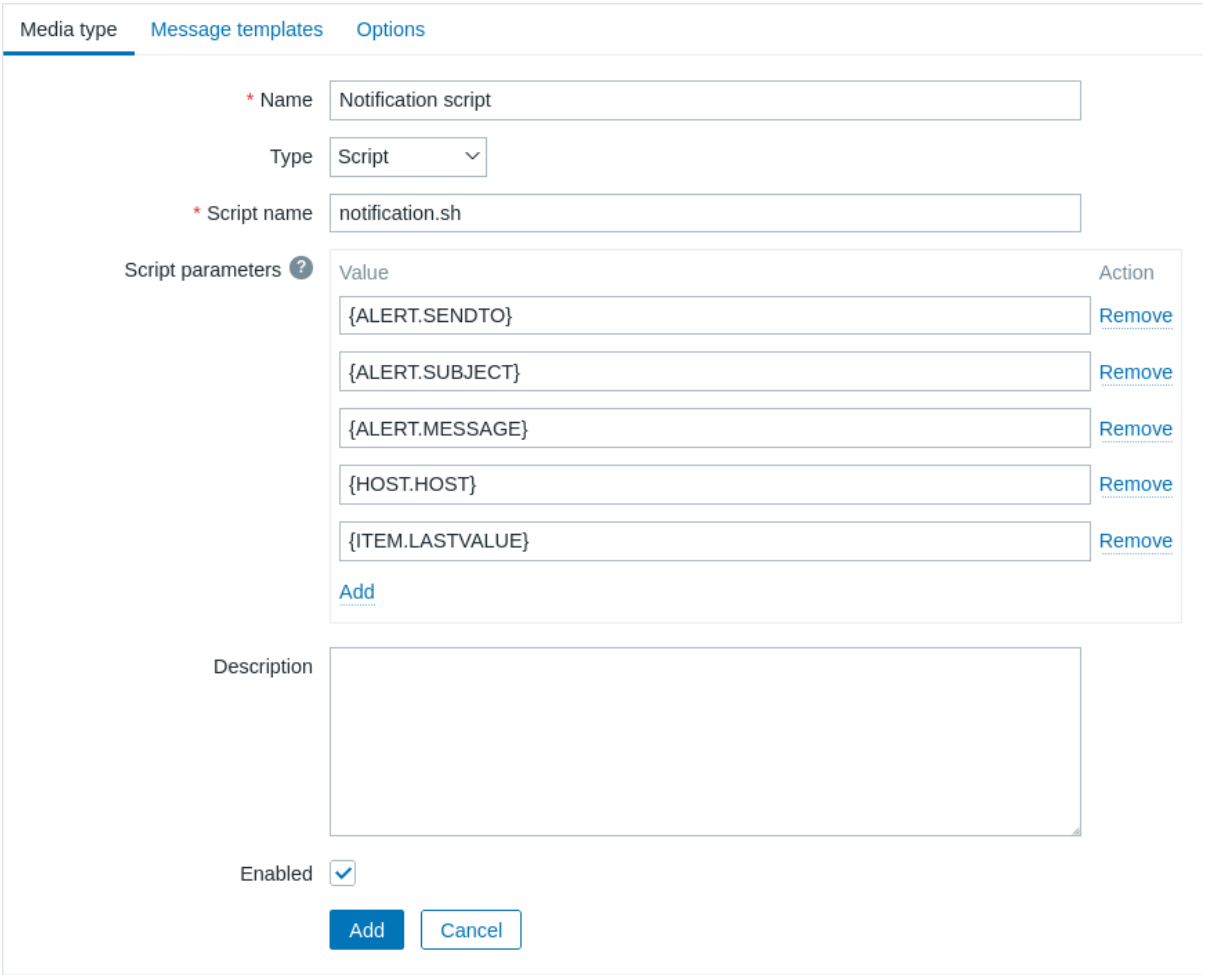
Переменные окружения для скрипта не сохраняются и не создаются, поэтому их следует обрабатывать явно.

Настройка

Чтобы настроить пользовательские скрипты оповещений как тип носителя:

1. Перейдите в раздел *Оповещения* > *Типы носителей*.
2. Нажмите *Создать тип носителя*.
3. Введите параметры типа носителя «Скрипт» в форме.

Вкладка **Тип носителя** содержит общие атрибуты типа носителя:



Все обязательные поля ввода отмечены красной звездочкой.

Следующие параметры относятся к типу носителя «Скрипт»:

Параметр	Описание
Имя скрипта	Введите имя файла скрипта (например, notification.sh), расположенного в каталоге, указанном в параметре конфигурации сервера AlertScriptsPath.
Параметры скрипта	Добавьте необязательные параметры скрипта, которые будут переданы скрипту в качестве аргументов командной строки в порядке их определения. Параметры скрипта поддерживают макросы {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} и все макросы, поддерживаемые в уведомлениях, а также макросы пользователей.

Подробнее о настройке сообщений по умолчанию и параметров обработки оповещений см. в разделе общие параметры типов носителей.

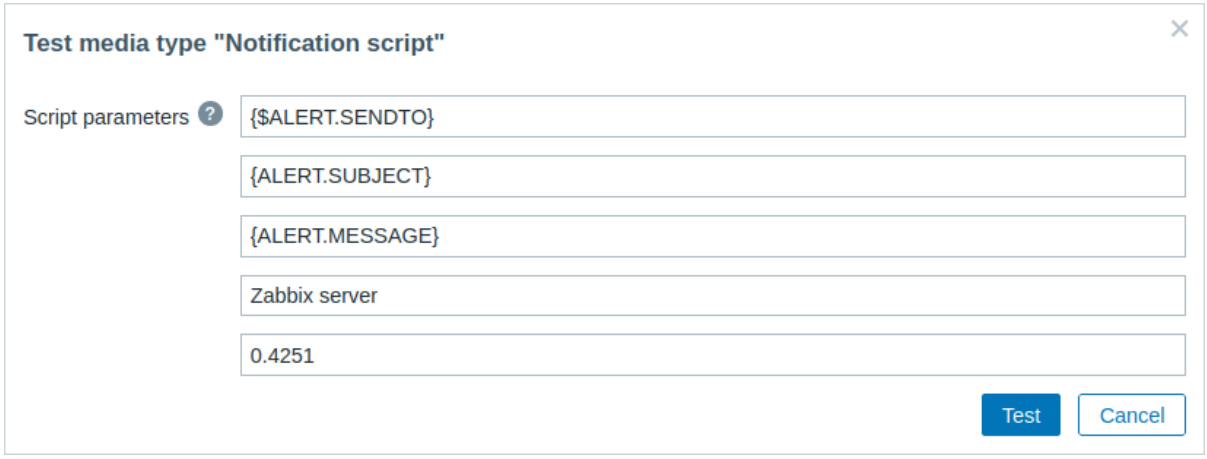
Warning:
Даже если скрипт оповещений не использует сообщения по умолчанию, шаблоны сообщений для типов операций, используемых этим типом носителя, все равно должны быть определены. В противном случае уведомление отправлено не будет.

Attention:
Если настроено несколько типов носителей «Скрипт», эти скрипты могут обрабатываться параллельно процессами обработки оповещений. Общее количество процессов обработки оповещений ограничено параметром `StartAlerters` в файле конфигурации сервера.

Тестирование

Чтобы протестировать настроенный тип медиа-элемента script:

1. Найдите нужный script в списке типов медиа-элементов.
2. Нажмите *Test* в последнем столбце списка; в всплывающем окне откроется форма тестирования. Форма тестирования будет содержать такое же количество параметров, которое настроено для типа медиа-элемента script.
3. При необходимости измените значения параметров script. Изменение влияет только на процедуру тестирования; фактические значения изменены не будут.
4. Нажмите *Test*.



Note:
При тестировании настроенного типа медиа-элемента script {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} и пользовательские макросы будут подставлены в свои значения, но макросы, связанные с событиями (например, {HOST.HOST}, {ITEM.LASTVALUE} и т. д.), подставлены не будут, поскольку во время тестирования нет связанного события, из которого можно получить эти сведения. Обратите внимание, что макросы внутри макросов {ALERT.SUBJECT} и {ALERT.MESSAGE} также не будут подставлены. Например, если значение {ALERT.SUBJECT} состоит из "Problem: {EVENT.NAME}", то макрос {EVENT.NAME} не будет подставлен.

Пользовательские способы оповещения

После настройки типа оповещения перейдите в раздел *Users* > *Users* и отредактируйте профиль пользователя, назначив этому пользователю данный тип оповещения. Шаги по настройке пользовательских способов оповещения, общие для всех типов оповещений, описаны на странице Типы оповещений.

Обратите внимание, что при определении пользовательского способа оповещения поле *Send to* не может быть пустым. Если это поле не используется в сценарии оповещения, введите любую комбинацию поддерживаемых символов, чтобы обойти требования проверки.

Вебхук

Обзор

Этот способ оповещений полезен для выполнения вызовов HTTP с использованием пользовательского кода JavaScript для прямой интеграции с внешними системами, такими как системы поддержки, чаты или мессенджеры. Вы можете выбрать — импортировать интеграцию, поставляемую Zabbix, или создать свою собственную интеграцию с нуля.

Интеграции

Доступны следующие интеграции, позволяющие использовать предопределенные типы медиа вебхуков для отправки уведомлений Zabbix в:

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
Помимо перечисленных здесь сервисов, Zabbix можно интегрировать со **Spiceworks** (вебхук не требуется). Чтобы преобразовать уведомления Zabbix в заявки Spiceworks, создайте тип медиа электронной почты и укажите адрес электронной почты службы поддержки Spiceworks (например, help@zabbix.on.spiceworks.com) в настройках профиля назначенного пользователя Zabbix.

Настройка

Чтобы начать использовать интеграцию с вебхуком:

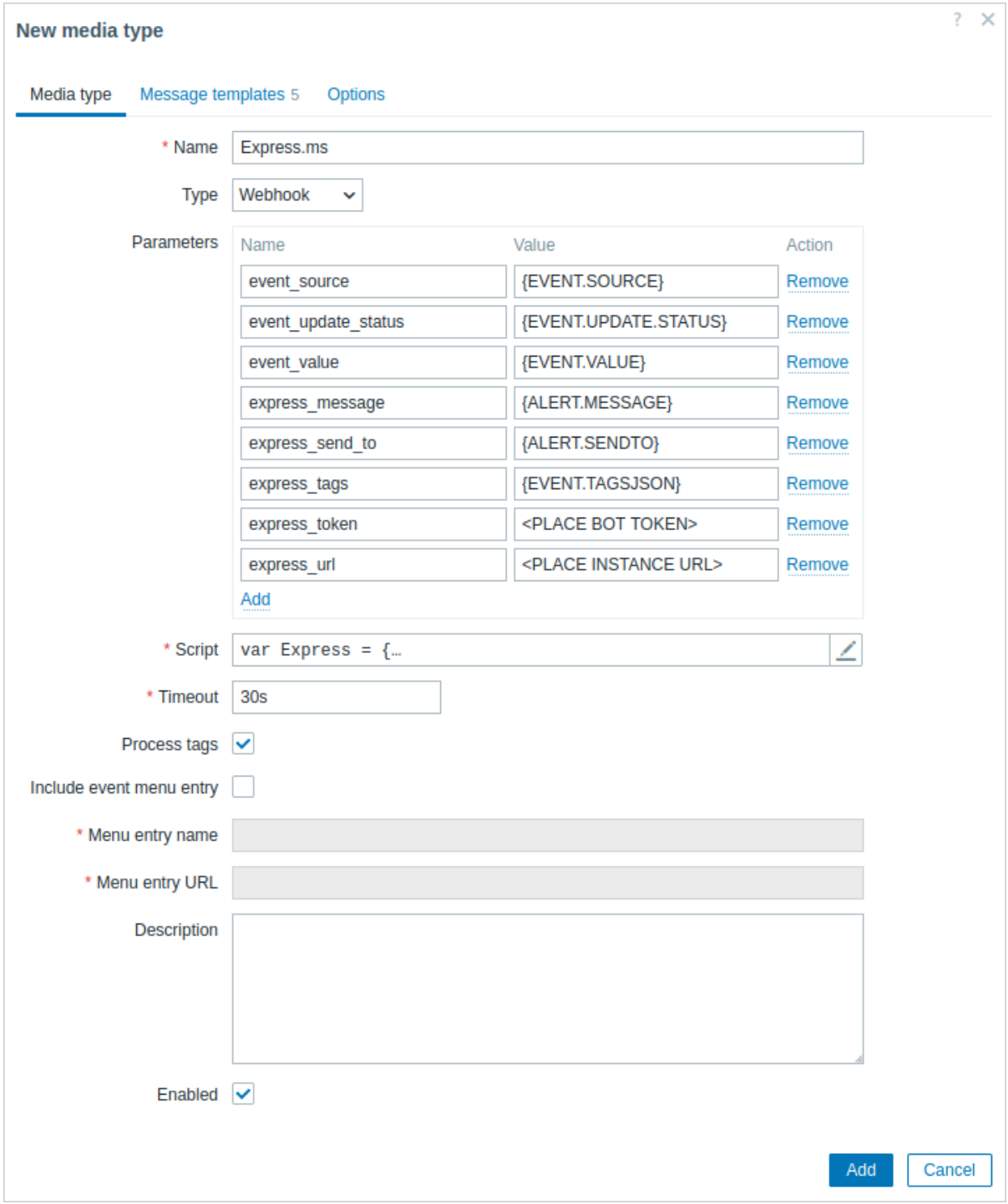
1. Найдите необходимый файл `.yaml` в каталоге `templates/media` загруженной версии Zabbix или скачайте его из репозитория Zabbix.

2. Импортируйте файл в установленный экземпляр Zabbix. Вебхук появится в списке типов носителей.
3. Настройте вебхук согласно инструкциям в файле `Readme.md` (для быстрого доступа к `Readme.md` можно нажать имя вебхука выше).

Чтобы создать пользовательский вебхук с нуля:

1. Перейдите в раздел *Оповещения* > *Типы носителей*.
2. Нажмите *Создать тип носителя*.
3. Введите параметры типа носителя вебхука в форме.

Вкладка **Тип носителя** содержит различные атрибуты, относящиеся к этому типу носителя:



Все обязательные поля ввода отмечены красной звездочкой.

Следующие параметры относятся к типу носителя вебхука:

Параметр	Описание
Параметры	Укажите переменные вебхука в виде пар атрибут-значение. Для предварительно настроенных вебхуков список параметров зависит от сервиса. Описание параметров см. в файле *Readme.md* вебхука. Для новых вебхуков по умолчанию включены несколько общих переменных (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); их можно оставить или удалить. Параметры вебхука поддерживают пользовательские макросы, все макросы, поддерживаемые в уведомлениях о проблемах, а также макросы {ALERT.SENDTO}, {ALERT.SUBJECT} и {ALERT.MESSAGE}. Если указан HTTP-прокси, поле поддерживает те же функции, что и поле HTTP-прокси в настройке элемента данных. Строка прокси может иметь префикс `[scheme]://`, указывающий тип используемого прокси (например, https, socks4, socks5; см. документацию).
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при нажатии в поле параметра или на значок карандаша рядом с ним. Этот код выполняет операцию вебхука. Скрипт представляет собой код функции, принимающей пары параметр-значение. Значения необходимо преобразовать в объекты JSON с помощью метода JSON.parse(), например: `var params = JSON.parse(value);`. Код имеет доступ ко всем параметрам, может выполнять запросы HTTP GET, POST, PUT и DELETE, поддерживать дополнительные методы, такие как CONNECT, PATCH, HEAD, OPTIONS и TRACE, а также управлять HTTP-заголовками и телом запроса. Скрипт должен содержать оператор return, иначе он будет недействительным. Он может возвращать статус OK с необязательным списком тегов и значений тегов (см. параметр *Обрабатывать теги*) или строку ошибки. События восстановления (созданные автоматически или в результате ручного закрытия) создаются сервером и содержат разрешенные теги событий (включая теги, унаследованные от шаблонов, узлов сети и триггеров). Скрипты вебхуков выполняются после создания оповещения, поэтому теги, возвращенные скриптом вебхука, добавляются только после первоначального создания оповещения и отсутствуют в макросах `{EVENT.TAGS}` и `{EVENT.RECOVERY.TAGS}` первоначального сообщения о проблеме или немедленного сообщения о восстановлении. *Примечание*: Рекомендуется использовать локальные переменные (например, `var local = 1`) вместо глобальных (например, `global = 1`), чтобы каждый скрипт работал со своими данными и не возникали конфликты между одновременными вызовами (см. известные проблемы). См. также: Рекомендации по разработке вебхуков, Примеры скриптов вебхуков, Дополнительные объекты JavaScript.
Тайм-аут	Тайм-аут выполнения JavaScript (1–60 с, по умолчанию 30 с). Поддерживаются суффиксы времени (например, `30s`, `1m`).
Обрабатывать теги	Установите флажок, чтобы обрабатывать значения свойств возвращаемого JSON как теги. Эти теги добавляются к существующим тегам проблемы. Обратите внимание: при использовании тегов вебхука вебхук должен возвращать объект JSON, содержащий как минимум пустой объект тегов: `var result = {tags: {}};` Примеры возвращаемых тегов: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Включить пункт меню события	Установите флажок, чтобы добавить в меню события пункт со ссылкой на созданную внешнюю заявку. Пункт будет добавлен для каждого включенного вебхука с установленным этим флажком. Обратите внимание: если параметры *Название пункта меню* и *URL пункта меню* содержат макросы {EVENT.TAGS.<tag name>}, пункт будет добавлен только в том случае, если эти макросы можно разрешить (то есть для события определены соответствующие теги). Если флажок установлен, вебхук не следует использовать для отправки уведомлений разным пользователям (вместо этого рассмотрите возможность создания выделенного пользователя) и не следует использовать в нескольких действиях оповещения для одного события проблемы.

Параметр	Описание
Название пункта меню	Укажите название пункта меню. Поддерживается макрос {EVENT.TAGS.<tag name>}. Это поле является обязательным только если установлен флажок *Включить пункт меню события*.
URL пункта меню	Укажите целевой URL пункта меню. Поддерживается макрос {EVENT.TAGS.<tag name>}. Это поле является обязательным только если установлен флажок *Включить пункт меню события*.

Подробные сведения о настройке сообщений по умолчанию и параметров обработки оповещений см. в разделе общие параметры типов носителей.

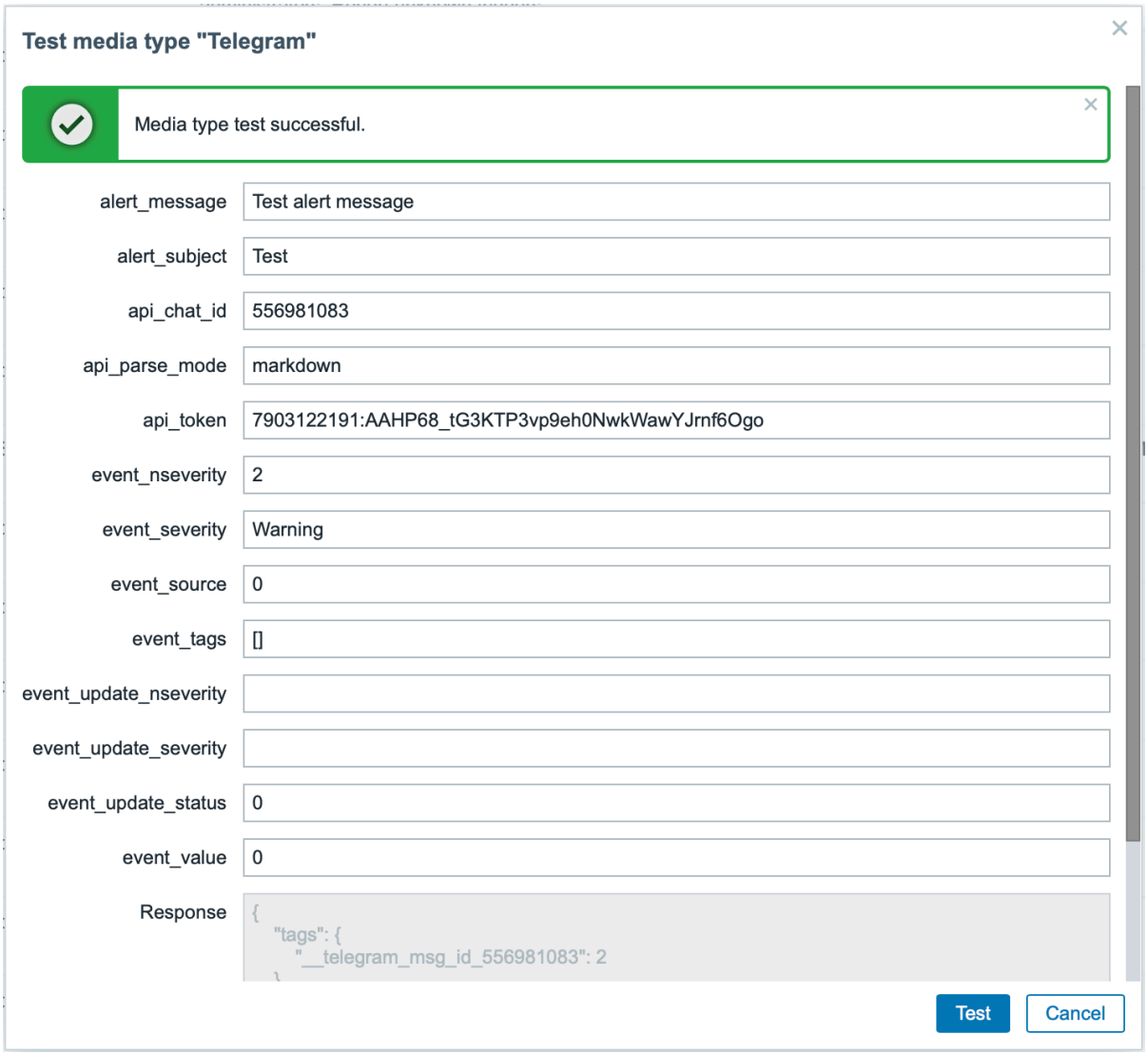
Warning:
Даже если вебхук не использует сообщения по умолчанию, шаблоны сообщений для типов операций, используемых этим вебхуком, все равно должны быть определены.

Тестирование

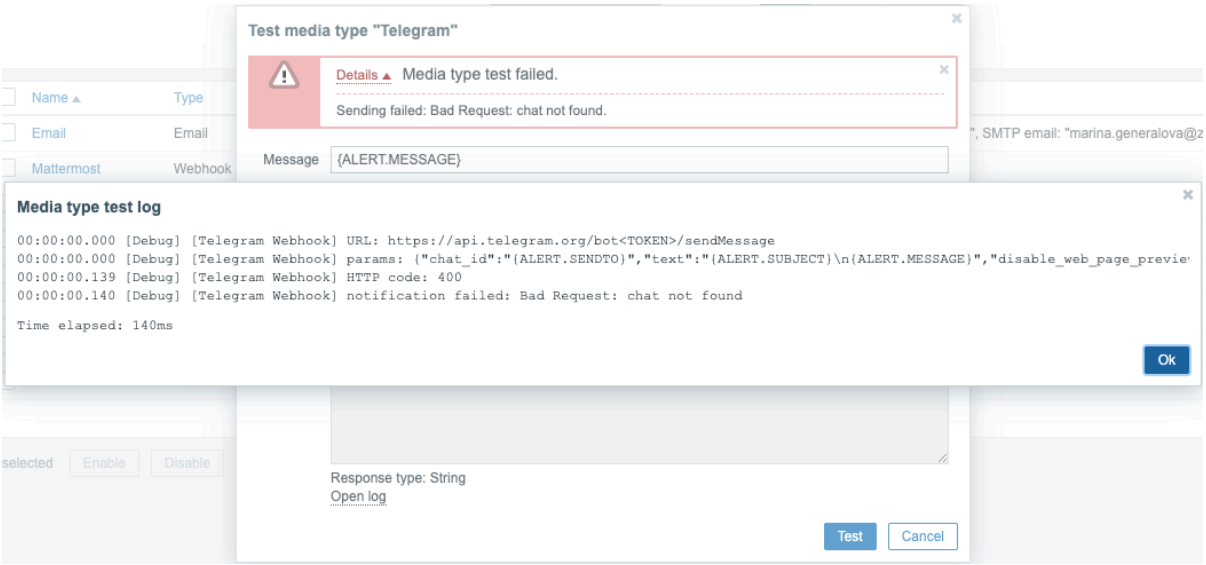
Чтобы протестировать настроенный тип носителя вебхука:

1. Найдите нужный вебхук в списке типов носителей.
2. Нажмите *Тест* в последнем столбце списка (откроется окно тестирования).
3. При необходимости измените значения параметров вебхука.
   Замените макросы примерами значений; в противном случае макросы не будут обработаны, и тест завершится ошибкой.
4. Нажмите *Тест*.

Замена или удаление значений в окне тестирования влияет только на процедуру тестирования, фактические значения атрибутов вебхука останутся без изменений.



Чтобы просмотреть записи журнала тестирования типа носителя, не закрывая окно тестирования, нажмите *Открыть журнал* (откроется новое всплывающее окно).



Если проверка вебхука успешна:

- Появится сообщение: *«Успешное тестирование способа оповещения (Media type test successful)»*.
- В сером поле *Ответ (Response)* отображается ответ сервера.
- Под полем *Ответ* указывается тип ответа («JSON» или «Строка (String)»).

Если проверка вебхука неуспешна:

- Отображается сообщение: *«Ошибка при тестировании способа оповещения (Media type test failed)»*, — с указанием

дополнительных сведений.

Пользовательский носитель

После настройки типа носителя перейдите в раздел *Пользователи* > *Пользователи* и назначьте носитель вебхука существующему пользователю или создайте нового пользователя, который будет представлять вебхук. Действия по настройке пользовательского носителя для существующего пользователя, общие для всех типов носителей, описаны на странице Типы носителей.

Если вебхук использует теги для хранения идентификатора тикета\сообщения, не назначайте один и тот же вебхук в качестве носителя разным пользователям, поскольку это может привести к ошибкам вебхука (относится к большинству вебхуков, использующих параметр *Include event menu entry*). В этом случае рекомендуется создать отдельного пользователя, который будет представлять вебхук:

1. После настройки типа носителя вебхука перейдите в раздел *Пользователи* > *Пользователи* и создайте отдельного пользователя Zabbix, который будет представлять вебхук, например пользователя с именем *Slack* для вебхука Slack. Все параметры, кроме носителя, можно оставить без изменений, поскольку этот пользователь не будет входить в Zabbix.
2. В профиле пользователя перейдите на вкладку *Носители* и добавьте вебхук, указав необходимые контактные данные. Если вебхук не использует поле *Send to*, введите любую комбинацию поддерживаемых символов, чтобы пройти проверку.
3. Предоставьте этому пользователю как минимум права на чтение узлов сети, для которых он должен отправлять оповещения.

При настройке действия оповещения добавьте этого пользователя в поле *Send to users* в разделе «Подробности операции» — это укажет Zabbix использовать вебхук для отправки уведомлений в рамках этого действия.

Настройка действий оповещения

Действия определяют, какие уведомления должны быть отправлены через вебхук. Шаги по настройке действий, использующих вебхуки, такие же, как и для всех остальных типов медиа, за исключением следующих случаев:

- Если вебхук использует теги вебхука для хранения ID тикета\сообщения и обработки операций обновления\разрешения, не используйте один и тот же вебхук в нескольких действиях оповещения для одного события проблемы. Если {EVENT.TAGS.<tag name>} существует и обновляется в вебхуке, его итоговое значение будет неопределенным. Чтобы избежать этого, используйте в вебхуке новое имя тега для хранения обновленных значений. Это относится к вебхукам Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad и Zendesk, предоставляемым Zabbix, а также к большинству вебхуков, использующих опцию *Include event menu entry*. Однако один вебхук можно использовать в нескольких операциях или шагах эскалации одного и того же действия, а также в разных действиях, которые не будут запущены одним и тем же событием проблемы из-за разных условий.
- При использовании вебхука в действиях для внутренних событий обязательно установите флажок *Custom message* и задайте пользовательское сообщение в конфигурации операции действия. В противном случае уведомление отправлено не будет.

Примеры скриптов вебхуков

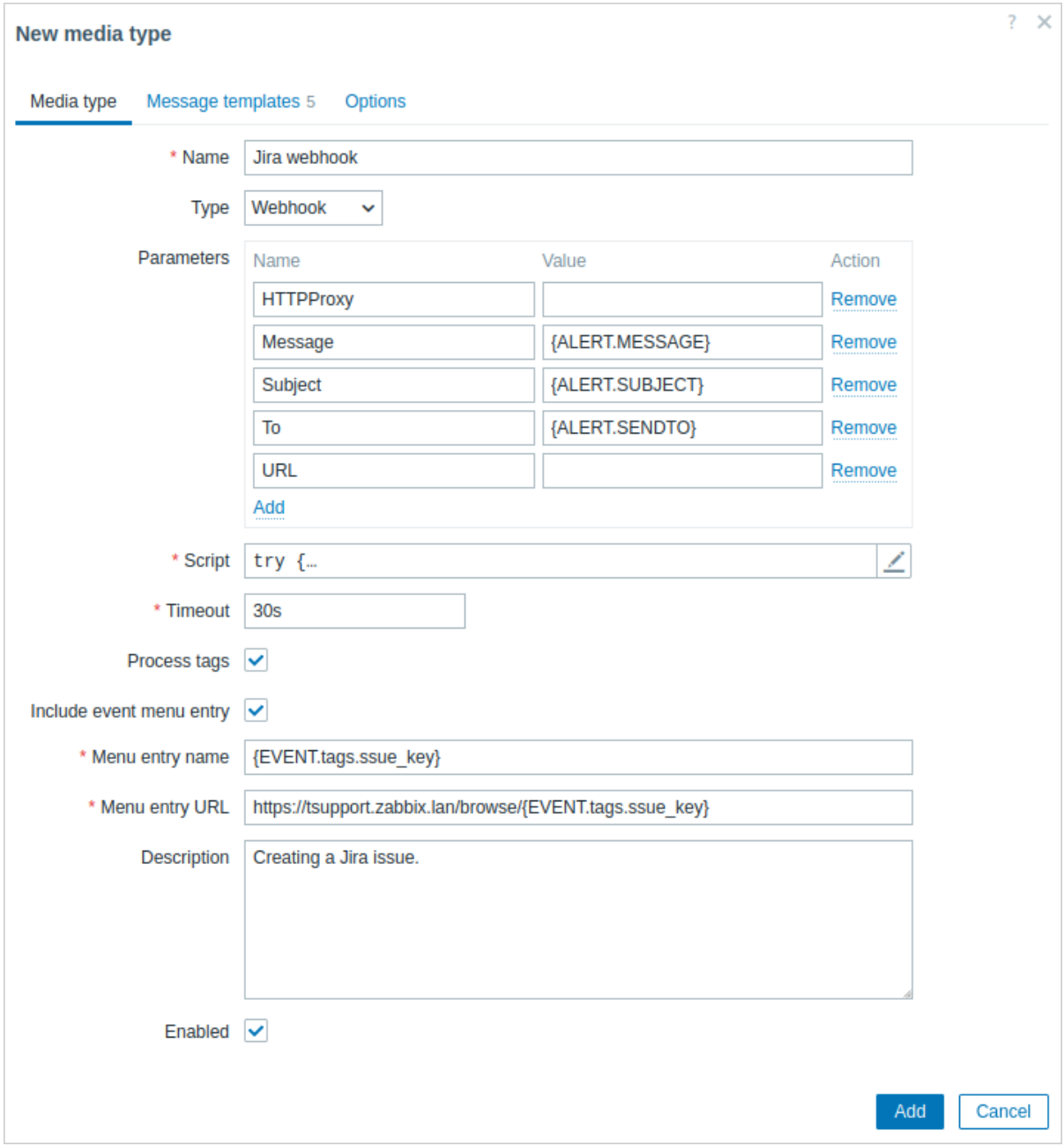
Обзор

Хотя Zabbix предлагает большое количество готовых интеграций с вебхуками, вы можете захотеть создать собственные вебхуки. В этом разделе приведены примеры пользовательских скриптов вебхуков (используемых в параметре *Script*). Описание других параметров вебхука см. в разделе вебхук.

> **Attention:**
> Не используйте не объявленные присваивания в JavaScript предварительной обработки. Используйте `var` для объявления локальных переменных.

События восстановления (независимо от того, сгенерированы ли они автоматически или после ручного закрытия) включают теги восстановленного события (в том числе теги, унаследованные от шаблонов, узлов сети и триггеров). Скрипты вебхуков выполняются после создания оповещения; поэтому теги, возвращаемые скриптом вебхука, применяются уже после первоначального создания оповещения и могут отсутствовать в оповещении, которое вызвало вебхук. Если для интеграции требуется, чтобы теги, созданные вебхуком, присутствовали в первоначальном уведомлении, получите теги события с сервера (например, с помощью API Event) или сохраняйте теги, созданные вебхуком, во внешнем постоянном хранилище и выполняйте сопоставление там.

Jira вебхук (пользовательский)



Этот скрипт создаст проблему (issue) в JIRA и вернёт некоторую информацию о созданной проблеме.

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);
```

```
    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

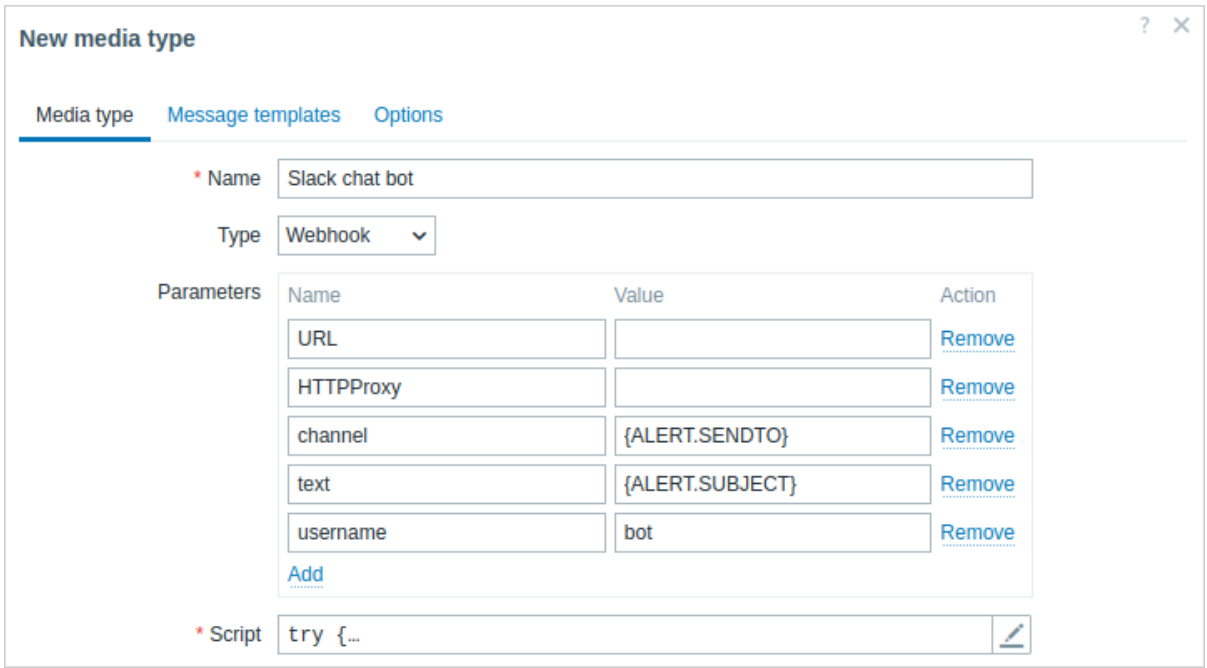
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Вебхук Slack (custom)

Этот вебхук будет пересылать уведомления из Zabbix в канал Slack.




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);
```

```
    response = req.post(params.URL, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

Действия

Обзор

Если вы хотите, чтобы в результате событий выполнялись какие-либо операции (например, отправлялись уведомления), необходимо настроить действия.

Действия можно определить в ответ на события всех поддерживаемых типов:

- Действия триггеров - для событий, когда состояние триггера изменяется между `OK` и `Problem`
- Действия сервисов - для событий, когда состояние сервиса изменяется между `OK` и `Problem`
- Действия обнаружения - для событий, когда выполняется обнаружение сети
- Действия авто-регистрации - для событий, когда новые активные агенты автоматически регистрируются (или изменяются метаданные узла сети для уже зарегистрированных)
- Внутренние действия - для событий, когда элементы данных становятся неподдерживаемыми или триггеры переходят в неизвестное состояние

Ключевые отличия действий сервисов:

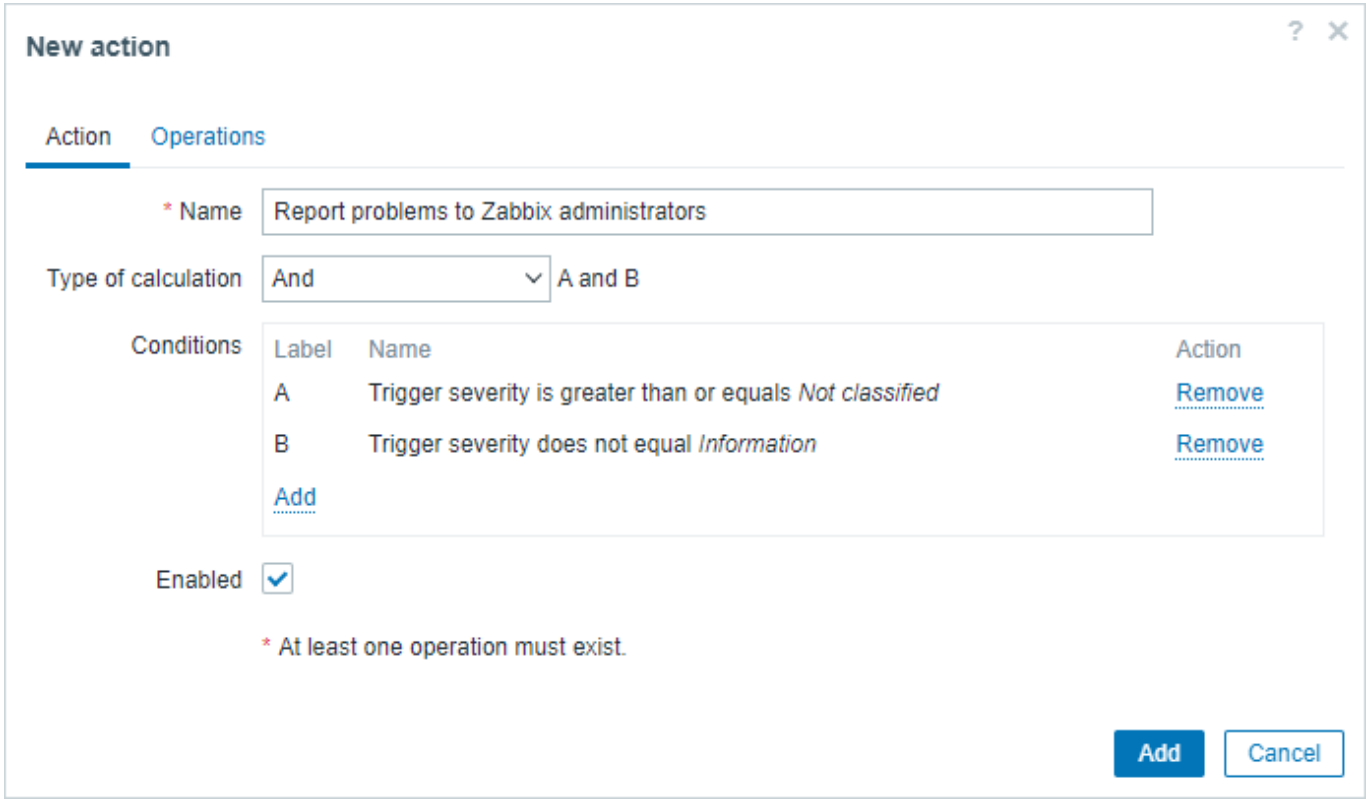
- Доступ пользователя к действиям сервисов зависит от прав доступа к сервисам, предоставленных ролью пользователя
- Действия сервисов поддерживают другой набор условий

Настройка действия

Чтобы настроить действие, выполните следующие шаги:

1. Перейдите в *Alerts* > *Actions* и выберите нужный тип действия в подменю (позже можно переключиться на другой тип, используя выпадающий список в заголовке).
2. Нажмите *Create action*.
3. Укажите имя действия.
4. Выберите conditions, при которых будут выполняться операции.
5. Выберите operations, которые нужно выполнить.

Общие атрибуты действия:



Все обязательные поля ввода отмечены красной звездочкой.

Parameter	Description
Name	Уникальное имя действия.
Type of calculation	Выберите option оценки условий действия (если условий больше одного): **And** - должны быть выполнены все условия. **Or** - достаточно, чтобы было выполнено одно условие. **And/Or** - комбинация двух вариантов: AND для разных типов условий и OR для условий одного типа. **Custom expression** - пользовательская формула вычисления для оценки условий действия.
Conditions	Список условий действия. Нажмите *Add*, чтобы добавить новое condition. Если условия не настроены, действие будет выполняться для каждого события, которое соответствует настраиваемому action type.
Enabled	Установите флажок, чтобы включить действие. В противном случае оно будет отключено.

Условия

Обзор

Можно задать, что действие будет выполнено, только если событие удовлетворяет определённому набору условий. Условия задаются при настройке действия.

Сопоставление условий чувствительно к регистру.

Действия на триггеры

В действиях на события триггеров можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Группа узлов сети (Host group)	равно не равно	Укажите группу узлов сети или исключение группы узлов сети. **равно** — событие относится к указанной группе узлов сети. **не равно** — событие не относится к указанной группе узлов сети. При указании родительской группы узлов сети косвенным образом будут выбраны все вложенные группы узлов сети. Чтобы выбрать только родительскую группу, все вложенные группы необходимо дополнительно перечислить с оператором **не равно**.
Шаблон (Template)	равно не равно	Укажите шаблон или исключение шаблона. **равно** — событие относится к триггеру, унаследованному из указанного шаблона. **не равно** — событие не относится к триггеру, унаследованному из указанного шаблона.
Узел сети (Tag name)	равно не равно	Укажите узел сети или исключение узла сети. **равно** — событие относится к указанному узлу сети. **не равно** — событие не относится к указанному узлу сети.
Имя тега (Tag name)	равно не равно содержит не содержит	Укажите тег события или исключение тега события. **равно** — событие имеет указанный тег. **не равно** — событие не имеет указанный тег. **содержит** — событие имеет тег, который содержит указанную строку. **не содержит** — событие не имеет тега, который содержит указанную строку.
Значение тега (Tag value)	равно не равно содержит не содержит	Укажите комбинацию тега события и его значения или исключение комбинации тега и значения. **равно** — событие имеет указанные тег и значение. **не равно** — событие не имеет указанные тег и значение. **содержит** — событие содержит указанные строки тега и значения. **не содержит** — событие не содержит указанные строки тега и значения.
Триггер (Trigger)	равно не равно	Укажите триггер или исключение триггера. **равно** — событие сгенерировано указанным триггером. **не равно** — событие сгенерировано любым триггером, отличным от указанного.
Имя события (Event name)	содержит не содержит	Укажите строку в имени события, сгенерированного триггером, или исключение строки. По умолчанию имя события совпадает с именем триггера, если в настройках триггера не было указано настраиваемого имени события. **содержит** — событие содержит в имени указанную строку. **не содержит** — указанная строка не содержится в имени события. Примечание: Заданное значение будет сравниваться с именем события со всеми раскрытыми макросами.
Важность триггера (Trigger severity)	равно не равно больше чем или равно меньше чем или равно	Укажите важность триггера. **равно** — совпадает с указанной важностью триггера. **не равно** — не совпадает с указанной важностью триггера. **больше чем или равно** — выше или равно указанной важности триггера. **меньше чем или равно** — ниже или равно указанной важности триггера.
Период времени (Time period)	в не в	Укажите период времени или исключение периода времени. **в** — время события в течение указанного периода времени. **не в** — время события не входит в указанный период времени. Смотрите страницу Спецификации периодов времени для получения более подробных сведений об этом формате. Поддерживаются пользовательские макросы.
Проблема подавлена (Problem is suppressed)	нет да	Укажите, если проблема подавлена (не отображается) по причине обслуживания узла сети. **нет** — проблема не подавлена. **да** — проблема подавлена.

Действия сервиса

В действиях сервиса можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Сервис	равно не равно	Укажите сервис или сервис, который нужно исключить. **равно** - событие относится к этому сервису. **не равно** - событие не относится к этому сервису. Указание родительского сервиса неявно выбирает все дочерние сервисы. Чтобы указать только родительский сервис, для всех вложенных сервисов дополнительно нужно задать оператор **не равно**.
Имя сервиса	содержит не содержит	Укажите строку в имени сервиса или строку, которую нужно исключить. **содержит** - событие сгенерировано сервисом, в имени которого содержится эта строка. **не содержит** - эта строка не найдена в имени сервиса.
Имя тега сервиса	равно не равно содержит не содержит	Укажите тег события или тег события, который нужно исключить. Теги событий сервиса можно определить в разделе конфигурации сервиса *Tags*. **равно** - событие имеет этот тег. **не равно** - событие не имеет этого тега. **содержит** - событие имеет тег, содержащий эту строку. **не содержит** - событие не имеет тега, содержащего эту строку.
Значение тега сервиса	равно не равно содержит не содержит	Укажите комбинацию тега события и значения или комбинацию тега и значения, которую нужно исключить. Теги событий сервиса можно определить в разделе конфигурации сервиса *Tags*. **равно** - событие имеет этот тег и значение. **не равно** - событие не имеет этого тега и значения. **содержит** - событие имеет тег и значение, содержащие эти строки. **не содержит** - событие не имеет тега и значения, содержащих эти строки.

Attention:
Обязательно определите шаблоны сообщений для действий сервиса в меню *Alerts* > *Media types*. В противном случае уведомления не будут отправляться.

Действия обнаружения

В событиях на основе обнаружения можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
IP-адрес узла сети	равно не равно	Укажите диапазон IP-адресов или диапазон, который необходимо исключить для обнаруженного узла сети. **равно** - IP-адрес узла сети входит в диапазон. **не равно** - IP-адрес узла сети не входит в диапазон. Можно использовать следующие форматы: Одиночный IP-адрес: 192.168.1.33 Диапазон IP-адресов: 192.168.1-10.1-254 Маска IP-адреса: 192.168.4.0/24 Список: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 В формате списка допускаются пробелы.

Тип условия	Поддерживаемые операторы	Описание
Тип сервиса	равно не равно	Укажите тип обнаруженного сервиса или тип сервиса, который необходимо исключить. **равно** - соответствует обнаруженному сервису. **не равно** - не соответствует обнаруженному сервису. Доступные типы сервисов: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, агент Zabbix, агент SNMPv1, агент SNMPv2, агент SNMPv3, эхо-запрос ICMP, telnet.
Порт сервиса	равно не равно	Укажите диапазон TCP-портов обнаруженного сервиса или диапазон, который необходимо исключить. **равно** - порт сервиса входит в диапазон. **не равно** - порт сервиса не входит в диапазон.
Правило обнаружения	равно не равно	Укажите правило обнаружения или правило обнаружения, которое необходимо исключить. **равно** - используется это правило обнаружения. **не равно** - используется любое другое правило обнаружения, кроме этого.
Проверка обнаружения	равно не равно	Укажите проверку обнаружения или проверку обнаружения, которую необходимо исключить. **равно** - используется эта проверка обнаружения. **не равно** - используется любая другая проверка обнаружения, кроме этой.
Объект обнаружения	равно	Укажите обнаруженный объект. **равно** - равен обнаруженному объекту (устройству или сервису).
Статус обнаружения	равно	**Доступен** - соответствует событиям «Узел сети доступен» и «Сервис доступен». **Недоступен** - соответствует событиям «Узел сети недоступен» и «Сервис недоступен». **Обнаружен** - соответствует событиям «Узел сети обнаружен» и «Сервис обнаружен». **Потерян** - соответствует событиям «Узел сети потерян» и «Сервис потерян».
Время доступности/недоступности	больше или равно меньше или равно	Время доступности для событий «Узел сети доступен» и «Сервис доступен». Время недоступности для событий «Узел сети недоступен» и «Сервис недоступен». **больше или равно** - больше или равно указанному значению. Параметр указывается в секундах. **меньше или равно** - меньше или равно указанному значению. Параметр указывается в секундах.
Полученное значение	равно не равно больше или равно меньше или равно содержит не содержит	Укажите значение, полученное от проверки агента (Zabbix, SNMP) в правиле обнаружения. Сравнение строк. Если для правила настроено несколько проверок агента Zabbix или SNMP, проверяются полученные значения для каждой из них (каждая проверка создает новое событие, которое сопоставляется со всеми условиями). **равно** - равно значению. **не равно** - не равно значению. **больше или равно** - больше или равно значению. **меньше или равно** - меньше или равно значению. **содержит** - содержит подстроку. Параметр указывается в виде строки. **не содержит** - не содержит подстроку. Параметр указывается в виде строки.
Прокси	равно не равно	Укажите прокси или прокси, который необходимо исключить. **равно** - используется этот прокси. **не равно** - используется любой другой прокси, кроме этого.
Группа прокси	равно не равно	Укажите группу прокси или группу прокси, которую необходимо исключить. **равно** - используется эта группа прокси. **не равно** - используется любая другая группа прокси, кроме этой.

Note:
Проверки сервисов в правиле обнаружения, результатом которых являются события обнаружения, не выполняются одновременно. Поэтому, если в действии настроено **несколько** значений для условий , или , они будут сравниваться с одним событием обнаружения за раз, но **не** с несколькими событиями одновременно. В результате действия с несколькими значениями для одних и тех же типов проверок могут выполняться некорректно.

Действия автоматической регистрации

В действиях на основе автоматической регистрации активного агента можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Метаданные узла сети	содержит не содержит соответствует не соответствует	Укажите метаданные узла сети или метаданные узла сети для исключения. **содержит** - метаданные узла сети содержат строку. **не содержит** - метаданные узла сети не содержат строку. Метаданные узла сети можно указать в файле конфигурации агента. **соответствует** - метаданные узла сети соответствуют регулярному выражению. **не соответствует** - метаданные узла сети не соответствуют регулярному выражению.
Имя узла сети	содержит не содержит соответствует не соответствует	Укажите имя узла сети или имя узла сети для исключения. **содержит** - имя узла сети содержит строку. **не содержит** - имя узла сети не содержит строку. **соответствует** - имя узла сети соответствует регулярному выражению. **не соответствует** - имя узла сети не соответствует регулярному выражению.
Прокси	равно не равно	Укажите прокси или прокси для исключения. **равно** - используется этот прокси. **не равно** - используется любой другой прокси (или сервер), кроме этого прокси.
Группа прокси	равно не равно	Укажите группу прокси или группу прокси для исключения. **равно** - используется эта группа прокси. **не равно** - используется любая другая группа прокси, кроме этой.

Действия на внутренние события

В действиях на внутренние события можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Тип события (Event type)	равно	**Элемент данных в состоянии ”не поддерживается” (Item in ”not supported” state)** — соответствует событию, когда элемент данных переходит из состояния «активировано» в состояние «не поддерживается». **Правило низкоуровневого обнаружения в состоянии ”не поддерживается” (Low-level discovery rule in ”not supported” state)** — соответствует событию, когда правило низкоуровневого обнаружения переходит из состояния «активировано» в состояние «не поддерживается». **Триггер в состоянии ”неизвестно” (Trigger in ”unknown” state)** — соответствует событию, когда триггер переходит из состояния «активировано» в состояние «неизвестно».
Группа узлов сети (Host group)	равно не равно	Укажите группу узлов сети или группу узлов сети для исключения. **равно** — событие относится к указанной группе узлов сети. **не равно** — событие не относится к указанной группе узлов сети.

Тип условия	Поддерживаемые операторы	Описание
Имя тега (Tag name)	равно не равно содержит не содержит	Укажите тег события или исключение тега события. **равно** — событие имеет указанный тег. **не равно** — событие не имеет указанный тег. **содержит** — событие имеет тег, который содержит указанную строку. **не содержит** — событие не имеет тега, который содержит указанную строку.
Значение тега (Tag value)	равно не равно содержит не содержит	Укажите комбинацию тега события и его значения или исключение комбинации тега и значения. **равно** — событие имеет указанные тег и значение. **не равно** — событие не имеет указанные тег и значение. **содержит** — событие содержит указанные строки тега и значения. **не содержит** — событие не содержит указанные строки тега и значения.
Шаблон (Template)	равно не равно	Укажите шаблон или шаблон для исключения. **равно** — событие относится к элементу данных / триггеру / правилу низкоуровневого обнаружения, унаследованному из указанного шаблона. **не равно** — событие не относится к элементу данных / триггеру / правилу низкоуровневого обнаружения, унаследованному из указанного шаблона.
Узел сети (Host)	равно не равно	Укажите узел сети или узел сети для исключения. **равно** — событие относится к указанному узлу сети. **не равно** — событие не относится к указанному узлу сети.

Тип вычисления

Для вычисления условий доступны следующие опции:

- **И (And)** — должны быть выполнены все условия

Обратите внимание, вычисление «И» запрещено использовать между несколькими триггерами, когда они добавлены с условием . Действия могут выполняться на основе события только одного триггера.

- **Или (Or)** — достаточно выполнения одного условия
- **И/Или (And/Or)** — комбинация из двух опций: И с различными типами условий и ИЛИ с одинаковым типом условий, например:

Группа узлов сети равно Oracle сервера
 Группа узлов сети равно MySQL сервера
 Имя события содержит «База данных не работает»
 Имя события содержит «База данных недоступна»

вычисляется как

(Группа узлов сети равно Oracle сервера **или** Группа узлов сети равно MySQL сервера**) и (**Имя события содержит «База данных не работает» **или** Имя события содержит «База данных недоступна»**)**

- **Пользовательское выражение** — формула вычисления, введенная пользователем, для оценки условий действия. Она должна включать в себя все условия (представленные в виде заглавных букв A, B, C, ...) и может включать пробелы, символы табуляции, скобки ( ), **and** (с учётом регистра), **or** (с учётом регистра), **not** (с учётом регистра).

Тогда как предыдущий пример с `/` был бы представлен в виде `(A or B) and (C or D)`, в пользовательском выражении вы также можете использовать несколько других методов вычисления:

(A and B) and (C or D)
 (A and B) or (C and D)
 ((A or B) and C) or D
 (not (A or B) and C) or not D
 и так далее.

Удаление объектов, используемых в действиях

При попытке удалить объекты, которые используются в условии/операции действия, пользователь получит соответствующее сообщение об ошибке.

Действия не отключаются, если условия или операции действия:

- содержат удаленные объекты;
- содержат объекты, которые больше не обнаруживаются низкоуровневым обнаружением и были удалены (по истечении периода *Удалить потерянные ресурсы*).

Условия/операции действия не удаляются, если в них отсутствуют объекты. Вместо этого удаленные объекты отображаются как *Удаленная группа узлов сети*, *Удаленный узел сети*, *Удаленный триггер* и т. д. В столбце "Info" списка действий отображается значок предупреждения с сообщением "This action has conditions or operations referencing deleted object(s)."

Операции

Обзор

Вы можете указать следующие операции по всем событиям:

- Отправка сообщений
- Выполнение удалённых команд

Attention:
Zabbix сервер не создаёт оповещения, если доступ к узлу сети явным образом «запрещён» пользователю, который задан в операции в качестве получателя, или если пользователь вовсе не имеет прав доступа, заданных к этому узлу сети.

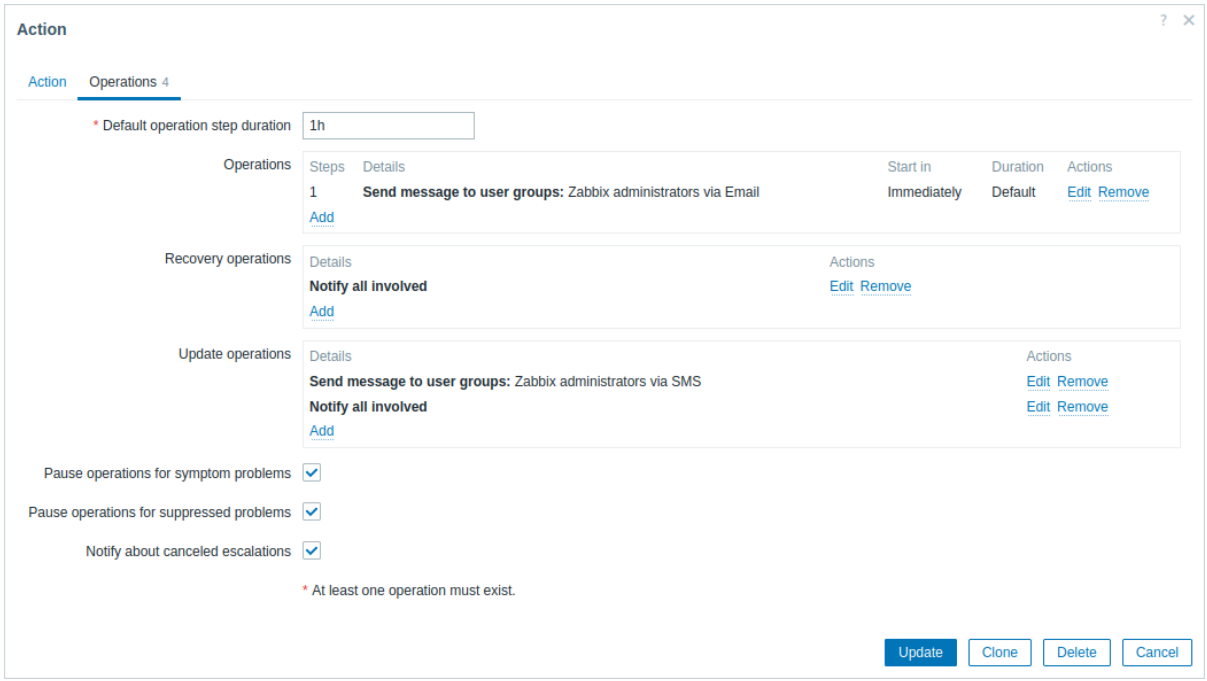
Для событий обнаружения и авторегистрации доступны дополнительные операции:

- Добавление узла сети
- Удаление узла сети
- Активация узла сети
- Деактивация узла сети
- Добавление в группу узлов сети
- Удаление из группы узлов сети
- Добавление тегов уровня узла сети
- Удаление тегов уровня узла сети
- Присоединение шаблона
- Отсоединение шаблона
- Установка режима инвентарных данных узлам сети

Настройка операции

Чтобы настроить операцию:

1. Перейдите на вкладку *Операции* в настройках действия.
2. Настройте параметры операции.



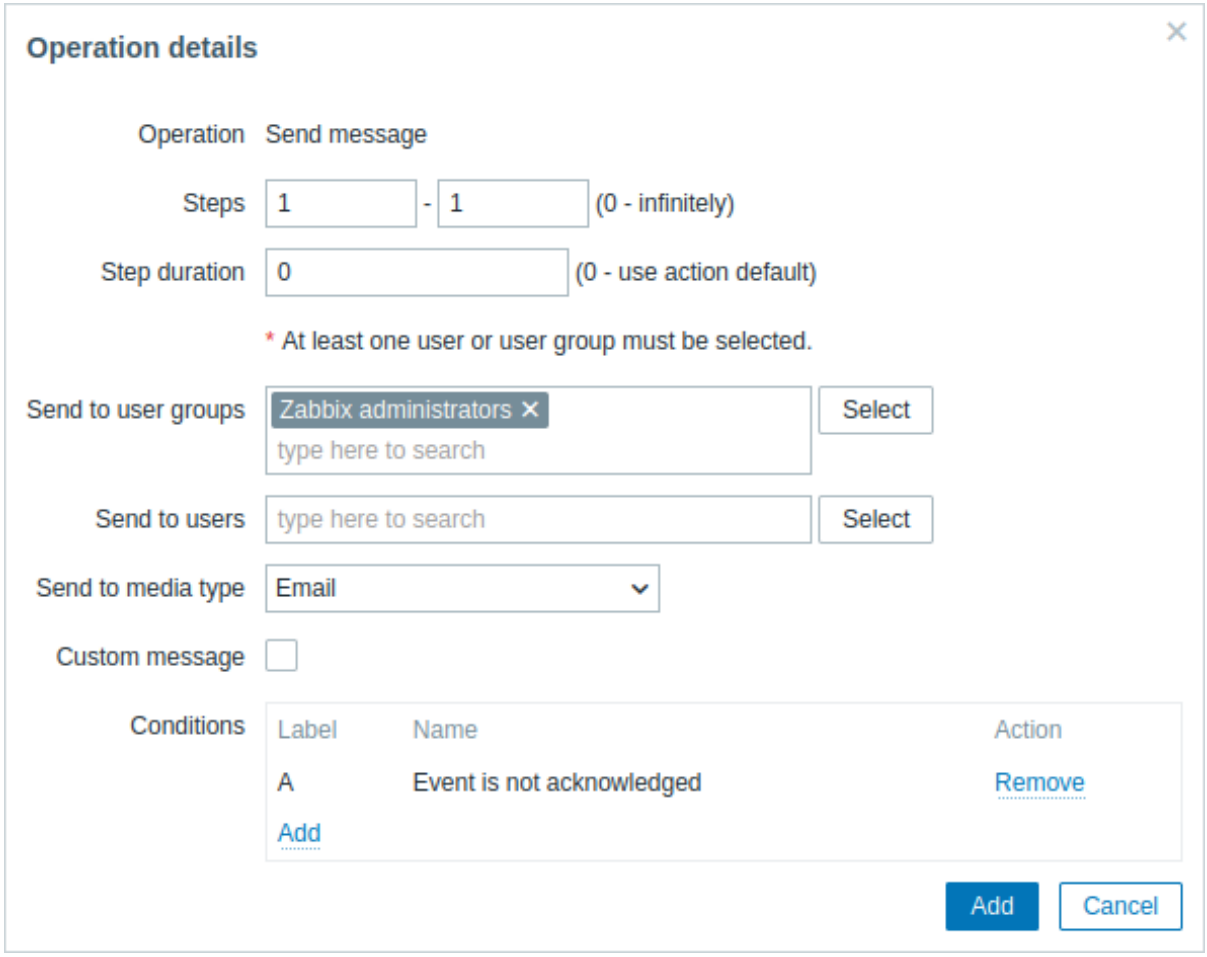
Общие атрибуты операции:

Параметр	Описание
Продолжительность шага операции по умолчанию	Продолжительность одного шага операции по умолчанию (от 60 секунд до 1 недели). Например, продолжительность шага в один час означает, что после выполнения операции до следующего шага пройдет один час. Поддерживаются суффиксы времени, например 60s, 1m, 2h, 1d. Поддерживаются пользовательские макросы.
Операции	Отображаются операции действия (если они есть) со следующими сведениями: **Шаги** - шаги эскалации, которым назначена операция. **Подробности** - тип операции и ее получатель/цель. В списке операций также отображаются используемый тип носителя (электронная почта, SMS или скрипт), а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. **Начало через** - время после события, через которое выполняется операция. **Продолжительность (сек.)** - отображается продолжительность шага. Если используется продолжительность по умолчанию, отображается значение *По умолчанию*, а при использовании пользовательской продолжительности - время. **Действия** - отображаются ссылки для редактирования и удаления операции.
Операции восстановления	Отображаются операции действия (если они есть) со следующими сведениями: **Подробности** - тип операции и ее получатель/цель. В списке операций также отображаются используемый тип носителя (электронная почта, SMS или скрипт), а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. **Действия** - отображаются ссылки для редактирования и удаления операции.
Операции обновления	Отображаются операции действия (если они есть) со следующими сведениями: **Подробности** - тип операции и ее получатель/цель. В списке операций также отображаются используемый тип носителя (электронная почта, SMS или скрипт), а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. **Действия** - отображаются ссылки для редактирования и удаления операции.
Приостановить операции для проблем с симптомами	Установите этот флажок, чтобы приостановить операции (после первой операции) для проблем с симптомами. Обратите внимание, что этот параметр влияет только на эскалации проблем; операции восстановления и обновления затронуты не будут. Этот параметр доступен только для *действий триггеров*.
Приостановить операции для подавленных проблем	Установите этот флажок, чтобы отложить начало операций на время периода обслуживания. После завершения обслуживания все операции будут выполнены, включая операции для событий, произошедших во время обслуживания. Обратите внимание, что этот параметр влияет только на эскалации проблем; операции восстановления и обновления затронуты не будут. Если снять этот флажок, операции будут выполняться без задержки даже во время периода обслуживания. Этот параметр недоступен для *действий сервисов*.
Уведомлять об отмененных эскалациях	Снимите этот флажок, чтобы отключить уведомления об отмененных эскалациях (если узел сети, элемент данных, триггер или действие отключены).

Все обязательные поля ввода отмечены красной звездочкой.

Чтобы настроить параметры новой операции, нажмите Add в блоке *Операции*. Чтобы отредактировать существующую операцию, нажмите Edit рядом с операцией. Откроется всплывающее окно, в котором можно изменить сведения о шаге операции.

Подробности операции



Параметр	Описание
Операция	Выберите операцию: **Отправить сообщение** - отправить сообщение пользователю. **<имя удаленной команды>** - выполнить удаленную команду. Команды доступны для выполнения, если они предварительно определены в глобальных скриптах с выбранной областью действия *Операция действия*. Дополнительные операции доступны для событий обнаружения и автоматической регистрации (см. выше).
Шаги	Выберите шаг или шаги, которым необходимо назначить операцию в расписании эскалации: **От** - выполнять начиная с этого шага. **До** - выполнять до этого шага (0=бесконечность, выполнение не ограничивается).
Продолжительность шага	Пользовательская продолжительность для этих шагов (0=использовать продолжительность шага по умолчанию). Поддерживаются суффиксы времени, например 60s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. Одному шагу можно назначить несколько операций. Если для этих операций указана разная продолжительность шага, учитывается и применяется к шагу наименьшая продолжительность.
Тип операции: отправка сообщения	
Отправить в группы пользователей	Выберите группы пользователей, которым необходимо отправить сообщение. Группа пользователей должна иметь как минимум права уровня «чтение» для узла сети, чтобы получать уведомления.
Отправить пользователям	Выберите пользователей, которым необходимо отправить сообщение. Пользователь должен иметь как минимум права уровня «чтение» для узла сети, чтобы получать уведомления.

Параметр	Описание
Отправить через тип носителя	Отправить сообщение через все доступные (настроенные и включенные) типы носителей или только через определенный тип.
Пользовательское сообщение	Если этот параметр выбран, можно настроить пользовательское сообщение. Для уведомлений о внутренних событиях через вебхуки пользовательское сообщение является обязательным.
Тема	Тема пользовательского сообщения. Тема может содержать макросы. Ограничение - 255 символов.
Сообщение	Пользовательское сообщение. Сообщение может содержать макросы. Количество символов ограничено и зависит от типа базы данных (дополнительную информацию см. в разделе Отправка сообщения).
Тип операции: удаленная команда	
Список целей	Выберите цели, на которых необходимо выполнить команду: **Текущий узел сети** - команда выполняется на узле сети, которому принадлежит триггер, вызвавший событие проблемы. Этот параметр не будет работать, если триггер содержит несколько узлов сети. **Узел сети** - выберите узел или узлы сети, на которых необходимо выполнить команду. **Группа узлов сети** - выберите группу или группы узлов сети, на которых необходимо выполнить команду. Указание родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети. Таким образом, удаленная команда также будет выполнена на узлах сети из вложенных групп. Команда на узле сети выполняется только один раз, даже если узел сети соответствует условию более одного раза (например, входит в несколько групп узлов сети, указан отдельно и через группу узлов сети). Список целей не имеет смысла, если пользовательский скрипт выполняется на сервере Zabbix. В этом случае выбор дополнительных целей приведет только к многократному выполнению скрипта на сервере. Обратите внимание, что для глобальных скриптов выбор цели также зависит от настройки *Группа узлов сети* в конфигурации глобального скрипта. Параметр *Список целей* недоступен для *действий сервисов*, поскольку в этом случае удаленные команды всегда выполняются на сервере Zabbix.
Условия	Условие выполнения операции: **Событие не подтверждено** - только если событие не подтверждено. **Событие подтверждено** - только если событие подтверждено. Параметр *Условия* доступен только для *действий триггеров*.

После завершения нажмите *Добавить*, чтобы добавить операцию в список *Операции*.

Отправка сообщения

Обзор

Отправка сообщений является одним из наилучших способов оповестить людей о наличии проблем. Вот по какой причине это одно из основных действий, предлагаемых Zabbix.

Настройка

Чтобы иметь возможность отправлять и получать уведомления из Zabbix, необходимо:

- Определить типы носителей, на которые будут отправляться сообщения.

Если операция выполняется за пределами периода Когда активен, определенного для выбранного типа носителя в настройках пользователя, сообщение отправлено не будет.

Степень важности триггера по умолчанию («Не классифицировано») **должна быть** выбрана в настройках типов носителей пользователя, если вы хотите получать уведомления о событиях, не связанных с триггерами, например о событиях обнаружения, активной автоматической регистрации агентов или внутренних событиях.

- Настроить операцию действия, которая отправляет сообщение на один из определенных типов носителей.

Attention:
Zabbix отправляет уведомления только тем пользователям, у которых есть как минимум права «чтение» на узел сети, сгенерировавший событие. Должен быть доступен как минимум один узел сети из выражения триггера.

Вы можете настроить пользовательские сценарии отправки сообщений с помощью эскалаций.

Для успешного получения и чтения электронных писем от Zabbix почтовые серверы/клиенты должны поддерживать стандартный формат «SMTP/MIME email», поскольку Zabbix отправляет данные в кодировке UTF-8 (если тема содержит только символы ASCII, она не кодируется в UTF-8). Тема и текст сообщения кодируются в base64 в соответствии со стандартом формата «SMTP/MIME email».

Ограничение на размер сообщения после подстановки всех макросов такое же, как ограничение на размер сообщения для удаленных команд.

Отслеживание сообщений

Вы можете просмотреть статус отправленных сообщений в *Monitoring* > *Problems*.

В столбце *Actions* вы можете увидеть сводную информацию о выполненных действиях. Зеленые числа обозначают отправленные сообщения, красные — неудачные сообщения. *In progress* означает, что действие инициировано. *Failed* сообщает, что ни одно действие не было успешно выполнено.

Если вы нажмете на время события, чтобы просмотреть подробности события, вы сможете увидеть сведения об отправленных (или не отправленных) сообщениях, связанных с событием, в блоке *Actions*.

В *Reports* > *Action log* вы увидите подробности всех выполненных действий для тех событий, для которых настроено действие.

Удаленные команды

Обзор

С помощью удалённых команд можно настроить автоматическое выполнение определённой заранее заданной команды на контролируемом узле сети при наступлении некоторого условия.

Таким образом, удалённые команды являются мощным механизмом интеллектуального проактивного мониторинга.

Наиболее очевидные варианты использования этой функции:

- автоматический перезапуск приложения (веб-сервера, промежуточного ПО, CRM), если оно не отвечает;
- использование команды IPMI «reboot» для перезагрузки удалённого сервера, если он не отвечает на запросы;
- автоматическое освобождение дискового пространства (удаление старых файлов, очистка `/tmp`) при его нехватке;
- перенос виртуальной машины с одного физического узла на другой в зависимости от загрузки ЦП;
- добавление новых узлов в облачную среду при недостатке ресурсов ЦП, диска, памяти или других ресурсов.

Настройка действия для удалённых команд аналогична настройке действия для отправки сообщения; единственное отличие заключается в том, что Zabbix выполняет команду вместо отправки сообщения.

Удалённые команды могут выполняться сервером Zabbix, прокси или агентом Zabbix. Обратите внимание на следующее:

- Удалённые команды на агенте Zabbix могут выполняться непосредственно сервером Zabbix или через прокси.
- Удалённые команды, выполняемые сервером Zabbix, запускаются в соответствии с описанием в разделе Выполнение команд, включая проверку кода возврата.
- Удалённые команды выполняются даже в том случае, если целевой узел сети находится в режиме обслуживания.

По умолчанию удалённые команды отключены как на агенте Zabbix, так и на прокси. Их можно включить следующим образом:

- Для удалённых команд, выполняемых агентом Zabbix, добавьте параметр `AllowKey=system.run[<command>,*]` или `AllowKeyRegexp=^system\.run\[<command>,.*\]$` для каждой разрешённой команды в конфигурации агента, где `*` соответствует любому режиму выполнения (`wait` или `nowait`).
- Для удалённых команд, выполняемых прокси, установите для параметра `EnableRemoteCommands` значение `1` в конфигурации прокси.

Обратите внимание, что включать удалённые команды на прокси не требуется, если они выполняются агентом Zabbix на узле сети, который контролируется прокси.

Attention:
Не используйте `AllowKey=system.run[*]` или `AllowKeyRegexp=^system\.run\[.*\]$` без необходимости, поскольку это разрешает выполнение любой команды на узле сети. В целях безопасности рекомендуется разрешать только точные команды, тогда как `system.run[*]` по умолчанию запрещён (`DenyKey=system.run[*]`).

См. также раздел Ограничение проверок агента.

Ограничение удаленной команды

Ограничение удаленной команды после разрешения всех макросов зависит от типа базы данных и набора символов (для хранения не-ASCII символов требуется более одного байта):

Database	Limit in characters	Limit in bytes
MySQL	65535	65535
PostgreSQL	65535	not limited
SQLite (only Zabbix proxy)	65535	not limited

Вывод выполнения удаленной команды (возвращаемое значение) ограничен 16 МБ (включая завершающие пробелы, которые усекаются). Ограничение удаленной команды IPMI зависит от установленной библиотеки IPMI. Обратите внимание, что ограничения базы данных применяются ко всем удаленным командам.

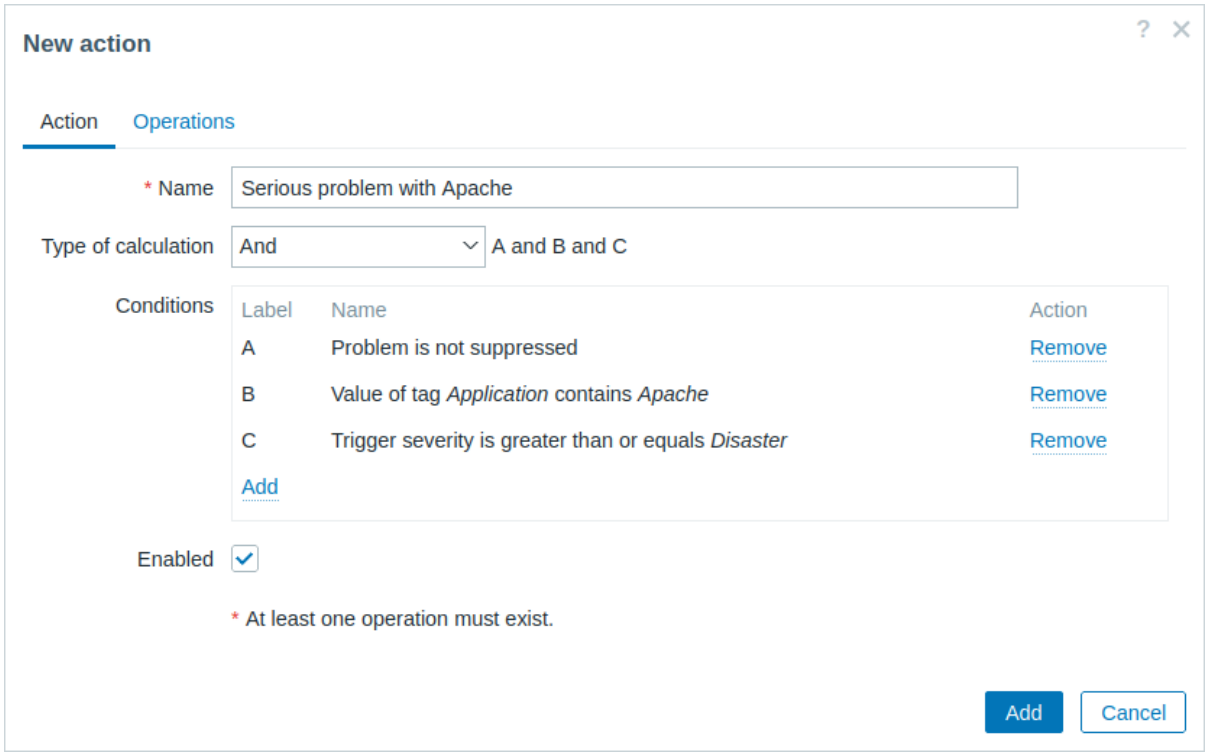
Настройка

Удаленные команды, выполняемые агентом Zabbix, сначала необходимо разрешить в конфигурации агента с помощью параметра `AllowKey` или `AllowKeyRegexp`.

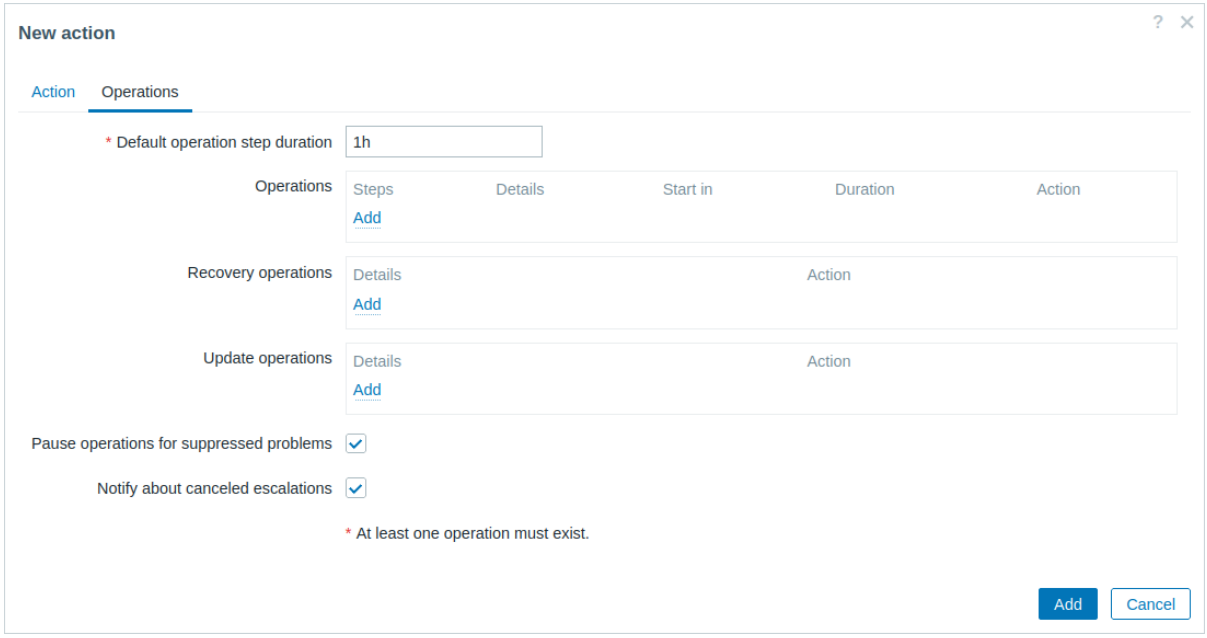
Убедитесь, что параметр `AllowKey=system.run[<command>,*]` или `AllowKeyRegexp=^system\.run\[<command>,.*\]$` добавлен в конфигурацию агента для каждой разрешенной команды, чтобы разрешить ее выполнение в режиме `nowait`. После изменения параметра перезапустите службу агента.

Затем при настройке нового действия в разделе *Оповещения* > *Действия* > *Действия с триггерами*:

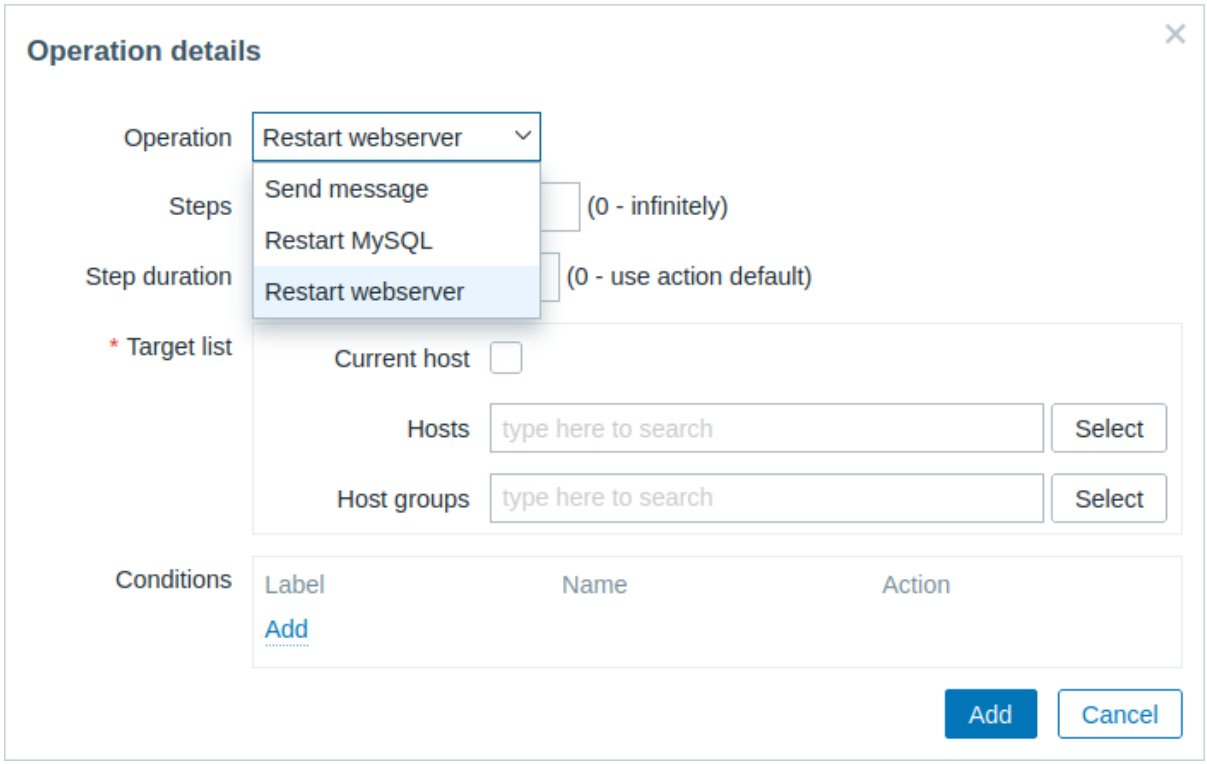
1. Определите соответствующие условия, например задайте активацию действия при возникновении любых проблем с уровнем серьезности «Катастрофа» в одном из приложений Apache.



2. На вкладке *Операции* нажмите *Добавить* в разделе *Операции*, *Операции восстановления* или *Операции обновления*.



3. Выберите один из предопределенных скриптов в раскрывающемся списке *Операция* и задайте *Список целей* для скрипта.



Предопределенные скрипты

Скрипты, доступные для операций действий (вебхук, скрипт, SSH, Telnet, IPMI), определяются в глобальных скриптах при выборе *Action operation* в качестве их области применения.

Например:

```
sudo /etc/init.d/apache restart
```

В этом случае Zabbix попытается перезапустить процесс Apache. При использовании этой команды убедитесь, что команда выполняется на агенте Zabbix (нажмите кнопку *Zabbix agent* напротив *Execute on*).

Attention:
Обратите внимание на использование `sudo` - у пользователя Zabbix по умолчанию нет прав на перезапуск системных служб. См. ниже рекомендации по настройке `sudo`.

Note:
Начиная с Zabbix agent 7.0, удаленные команды также могут выполняться на агенте, работающем в активном режиме. Zabbix agent — в активном или пассивном режиме — должен работать на удаленном узле сети и выполняет команды в фоновом режиме.

Удаленные команды на агенте Zabbix выполняются без тайм-аута с помощью ключа `system.run[,nowait]` и не проверяются на результаты выполнения. На сервере Zabbix и прокси Zabbix удаленные команды выполняются с тайм-аутом, заданным в параметре `TrapperTimeout` файла `zabbix_server.conf` или `zabbix_proxy.conf`, и проверяются на результаты выполнения. Дополнительную информацию см. в разделе *Script timeout*.

Права доступа

Убедитесь, что пользователь `zabbix` имеет права на выполнение настроенных команд. Возможно, имеет смысл использовать `sudo`, чтобы предоставить доступ к привилегированным командам. Чтобы настроить доступ, выполните от имени root:

```
visudo
```

Примеры строк, которые можно использовать в файле `sudoers`:

```
#              'zabbix'                    .
zabbix ALL=NOPASSWD: ALL

#              'zabbix'         apache       .
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
В некоторых системах файл `sudoers` не позволит удаленным пользователям выполнять команды. Чтобы изменить это, закомментируйте параметр `requiretty` в `/etc/sudoers`.

Удалённые команды при наличии нескольких интерфейсов

Если целевая система имеет несколько интерфейсов выбранного типа (Zabbix агент или IPMI), удалённые команды будут выполнены с использованием интерфейса по умолчанию.

Имеется возможность выполнения удалённых команд через SSH и Telnet с использованием другого интерфейса, отличного от интерфейса Zabbix агента. Доступный интерфейс, который будет использоваться, выбирается в следующем порядке:

- Zabbix агент интерфейс по умолчанию
- SNMP интерфейс по умолчанию
- JMX интерфейс по умолчанию
- IPMI интерфейс по умолчанию

Удалённые команды через IPMI

В случае использования удалённых команд через IPMI нужно использовать следующий синтаксис:

```
<     > [<     >]
```

где

- <команда> — одна из IPMI команд без пробелов
- <значение> — «on», «off» или любое неотрицательное целое число. <значение> — опциональный параметр.

Примеры

Примеры глобальных скриптов, которые можно использовать как удалённые команды в операциях к действиям.

Пример 1

Перезапуск Windows при определенном условии.

Чтобы автоматически перезапустить Windows при обнаружении проблемы Zabbix, определите следующий скрипт:

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`c:\windows\system32\shutdown.exe -r -f`

Пример 2

Перезагрузка узла сети с использованием управления по IPMI.

Параметр скрипта	Значение
Область	«Операция действия»
Тип	«IPMI»
Команда	reset

Пример 3

Выключение узла сети с использованием управления по IPMI.

Параметр скрипта	Значение
Область	«Операция действия»
Тип	«IPMI»
Команда	power off

Дополнительные операции

Обзор

В этом разделе вы можете найти некоторые сведения о дополнительных операциях на события обнаружения / автоматической регистрации.

Добавление узла сети

Узлы сети добавляются в процессе обнаружения, сразу после того, как узел сети обнаружен, а не в конце процесса обнаружения.

Note:
Поскольку обнаружение сети может занять некоторое время из-за большого количества недоступных узлов сети/служб, рекомендуется проявлять терпение и использовать разумные диапазоны IP-адресов.

При добавлении узла сети его имя определяется стандартной функцией **gethostbyname**. Если узел сети удается разрешить, используется разрешенное имя. Если нет, используется IP-адрес. Кроме того, если для имени узла сети необходимо использовать IPv6-адрес, все символы ":" (двоеточия) заменяются на "_" (подчеркивания), поскольку двоеточия не допускаются в именах узлов сети.

Attention:
Если обнаружение выполняется через прокси, в настоящее время поиск имени узла все равно выполняется на сервере Zabbix.

Attention:
Если узел сети уже существует в конфигурации Zabbix с тем же именем, что и только что обнаруженный, Zabbix добавит **_N** к имени узла сети, где **N** — возрастающее число, начиная с 2.

Использование макросов в сообщениях

Обзор

Для более эффективного оповещения о проблемах можно в теме и в тексте сообщений использовать макросы.

В дополнение к встроенным макросам, также поддерживаются пользовательские макросы и макросы выражений. Доступен полный список макросов, поддерживаемых Zabbix.

Примеры

Приведённые здесь примеры иллюстрируют то, как вы можете использовать макросы в сообщениях.

Пример 1

Тема сообщения:

```
    : {TRIGGER.NAME}
```

При получении сообщения тема этого сообщения будет заменена на нечто подобное:

```
    :                          Zabbix
```

Пример 2

Сообщение:

```
          : {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

При получении сообщения само сообщение будет заменено на нечто подобное:

```
          : 1.45
```

Пример 3

Сообщение:

```
          : {?last(/{HOST.HOST}/{ITEM.KEY})}
     15    : {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
     15    : {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

При получении сообщения само сообщение будет заменено на нечто подобное:

```
          : 1.45
     15    : 2.33
     15    : 1.01
```

Пример 4

Сообщение:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Полученное сообщение будет содержать ссылку на страницу *Детали события*, которая предоставит информацию о событии, его триггере и списке последних событий, сгенерированных этим триггером.

Пример 5

Информирование о значениях с нескольких узлов сети из выражения триггера.

Сообщение:

```
       : {TRIGGER.NAME}
          : {TRIGGER.EXPRESSION}

1.                  {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2.                  {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

При получении сообщения само сообщение будет заменено на нечто подобное:

```
       :
          : last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[percpu,avg1]

1.                  Myhost: 0.83 (           (         1         ))
2.                  Myotherhost: 5.125 (           (         1         ))
```

Пример 6

Получение деталей по событию о проблеме и событию о восстановлении в сообщении о восстановлении:

Сообщение:

```
    :

ID     : {EVENT.ID}
         : {EVENT.VALUE}
          : {EVENT.STATUS}
        : {EVENT.TIME}
       : {EVENT.DATE}
         : {EVENT.AGE}
            : {EVENT.ACK.STATUS}
               : {EVENT.UPDATE.HISTORY}
```

```
        :

ID      : {EVENT.RECOVERY.ID}
          : {EVENT.RECOVERY.VALUE}
           : {EVENT.RECOVERY.STATUS}
         : {EVENT.RECOVERY.TIME}
        : {EVENT.RECOVERY.DATE}
           : {EVENT.OPDATA}
```

При получении сообщения эти макросы будет заменены на нечто подобное:

```
     :

ID      : 21874
          : 1
           : PROBLEM
         : 13:04:30
        : 2018.01.02
          : 5m 0s
             : Yes
                : 2018.01.02 13:05:51 "John Smith (Admin)"
     :        .

        :

ID      : 21896
          : 0
           : OK
         : 13:10:07
        : 2014.01.02
           :          : 0.83
```

Операции восстановления

Обзор

Операции восстановления позволяют вам получить оповещения при решении проблем.

В операциях восстановления поддерживаются как сообщения, так и удалённые команды. Хоть можно добавить несколько операций, эскалации не поддерживаются — все операции назначаются на единственный шаг, поэтому будут выполняться одновременно.

Сценарии применения

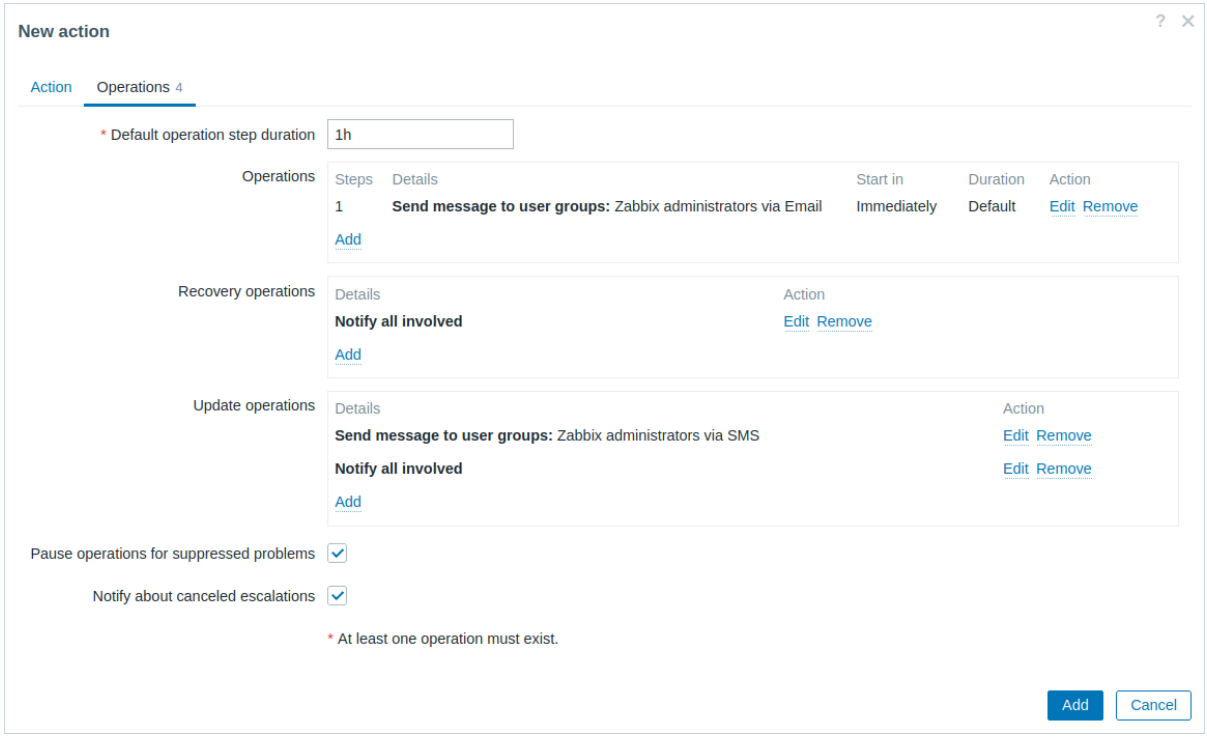
Некоторые сценарии использования операций восстановления заключаются в следующем:

1. Оповещение о восстановлении всех пользователей, которые были ранее оповещены о проблеме:
   - Выберите в качестве типа операции *Оповещать всех участников (Notify all involved)*
2. Наличие нескольких операций по восстановлению: отправка оповещений и выполнение удалённой команды:
   - Добавьте типы операций для отправки сообщения и выполнения команды
3. Открытие задачи во внешней системе поддержки клиентов / системе задач и закрытие этих задач при решении проблемы:
   - Создайте внешний скрипт, который будет взаимодействовать с системой поддержки клиентов.
   - Создайте действие, имеющее операцию, которая выполняет этот скрипт и таким образом открывает задачу.
   - Добавьте операцию восстановления, которая выполнит этот скрипт с другими параметрами и закроет задачу.
   - Используйте макрос {EVENT.ID} в качестве ссылки на исходную проблему.

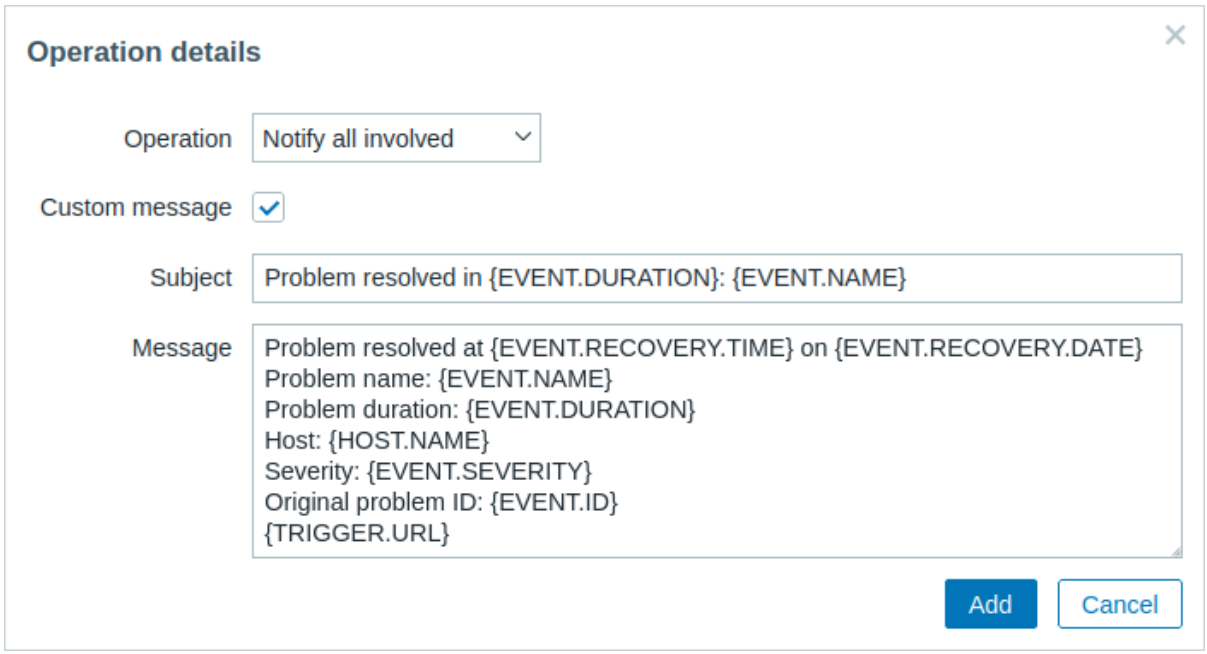
Настройка операции восстановления

Для настройки операции восстановления используйте следующие шаги:

1. Перейдите на вкладку *Операции (Operations)* в настройках действия.
2. Чтобы настроить новую операцию восстановления, нажмите *Добавить (Add)* в блоке *Операции восстановления (Recovery operations)*. Чтобы изменить существующую операцию, нажмите *Изменить (Edit)* сразу после операции.
3. Настройте подробности операции.



Детали операции восстановления



Для событий восстановления доступно три типа операций:

- **Отправить сообщение (Send message)** — отправка сообщения о восстановлении указанному пользователю;
- **Оповещать всех участников (Notify all involved)** — отправка сообщения всем пользователям, кто ранее получал оповещение о событии о проблеме;
- **<имя удалённой команды>** — выполнение удалённой команды. Команды доступны для выполнения, если предварительно были добавлены в глобальные скрипты с опцией *Операция действия (Action operation)*, выбранной в качестве области применения.

Параметры по каждому типу операции описаны ниже. Все обязательные поля ввода отмечены красной звёздочкой. Когда завершите, нажмите *Добавить (Add)* для добавления операции в список *Операции восстановления (Recovery operations)*.

Note:
Обратите внимание, что если один и тот же получатель указан в нескольких типах операций без указания *Пользовательское сообщение (Custom message)*, то дубликаты оповещений не отсылаются.

Тип операции: отправить сообщение

Параметр	Описание
Отправить группам пользователей	Выберите группы пользователей, которым нужно отправить сообщение о восстановлении. Группа пользователей должна иметь как минимум права "read" permissions на узел сети, чтобы получить уведомление.
Отправить пользователям	Выберите пользователей, которым нужно отправить сообщение о восстановлении. Пользователь должен иметь как минимум права "read" permissions на узел сети, чтобы получить уведомление.
Отправить через тип медиа	Отправить стандартное сообщение о восстановлении всем доступным (настроенным и включенным) типам медиа или только определенному.
Пользовательское сообщение	Если выбрано, можно определить пользовательское сообщение.
Тема	Тема пользовательского сообщения. Тема может содержать макросы.
Сообщение	Пользовательское сообщение. Сообщение может содержать макросы.

Тип операции: удаленная команда

Параметр	Описание
Список целей	Выберите цели, на которых необходимо выполнить команду: **Текущий узел сети** — команда выполняется на узле сети с триггером, вызвавшим событие проблемы. Этот вариант не будет работать, если в триггере указано несколько узлов сети. **Узел сети** — выберите узел или узлы сети, на которых необходимо выполнить команду. **Группа узлов сети** — выберите группу или группы узлов сети, на которых необходимо выполнить команду. Указание родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети. Таким образом, удаленная команда также будет выполнена на узлах сети из вложенных групп. Команда на узле сети выполняется только один раз, даже если узел сети соответствует условию более одного раза (например, входит в несколько групп узлов сети, указан отдельно и входит в группу узлов сети). Список целей не имеет смысла, если команда выполняется на сервере Zabbix. В этом случае выбор дополнительных целей приведет только к многократному выполнению команды на сервере. Обратите внимание, что для глобальных скриптов выбор цели также зависит от настройки *Группа узлов сети* в конфигурации глобального скрипта.

Тип операции: оповещать всех участников

Параметр	Описание
Пользовательское сообщение (Custom message)	Если выбрано, будет использовано пользовательское сообщение.
Тема (Subject)	Тема пользовательского сообщения. Тема может содержать макросы.
Сообщение (Message)	Пользовательское сообщение. Сообщение может содержать макросы.

Операции обновления

Обзор

Операции обновления доступны в действиях со следующими источниками событий:

- *Триггеры* - когда проблемы обновляются другими пользователями, то есть при добавлении комментария, подтверждении, изменении уровня серьезности, закрытии (вручную);
- *Сервисы* - когда уровень серьезности сервиса изменился, но сервис все еще не восстановлен.

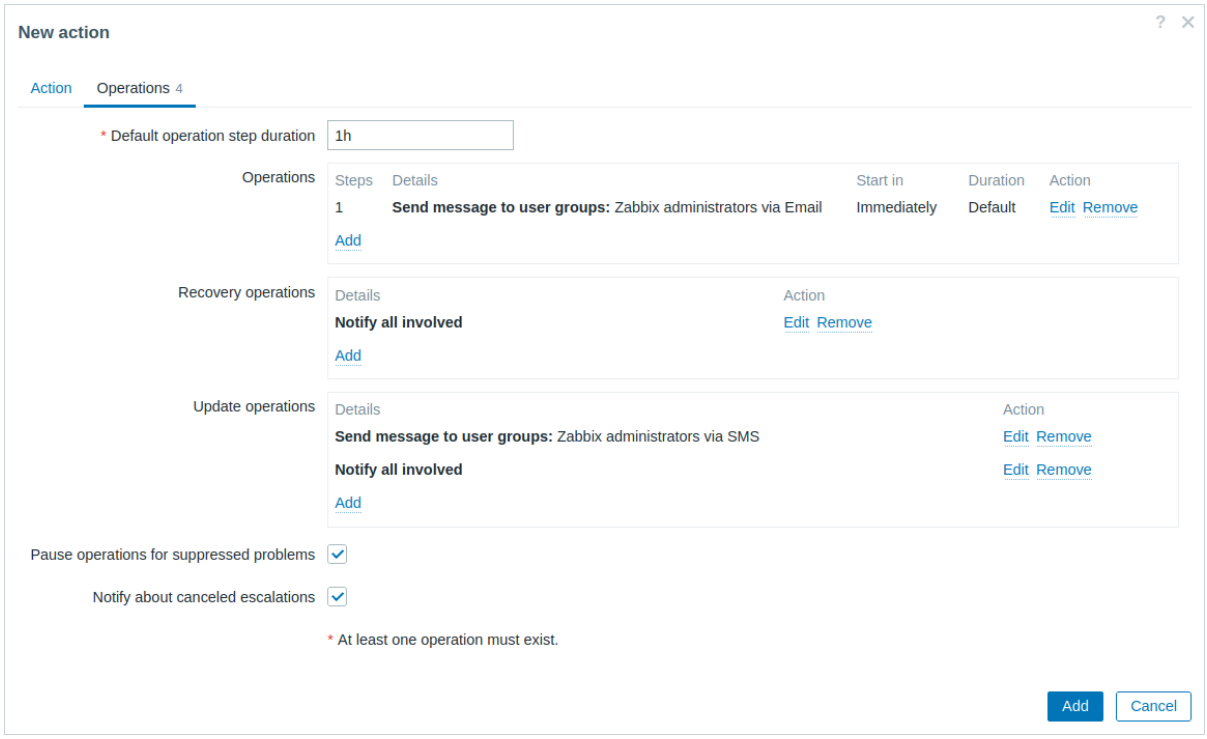
Обратите внимание, что пользователи не получают уведомления о своих собственных изменениях.

В операциях обновления поддерживаются как сообщения, так и удаленные команды. Хотя можно добавить несколько операций, эскалация не поддерживается - все операции назначаются одному шагу и, следовательно, выполняются одновременно.

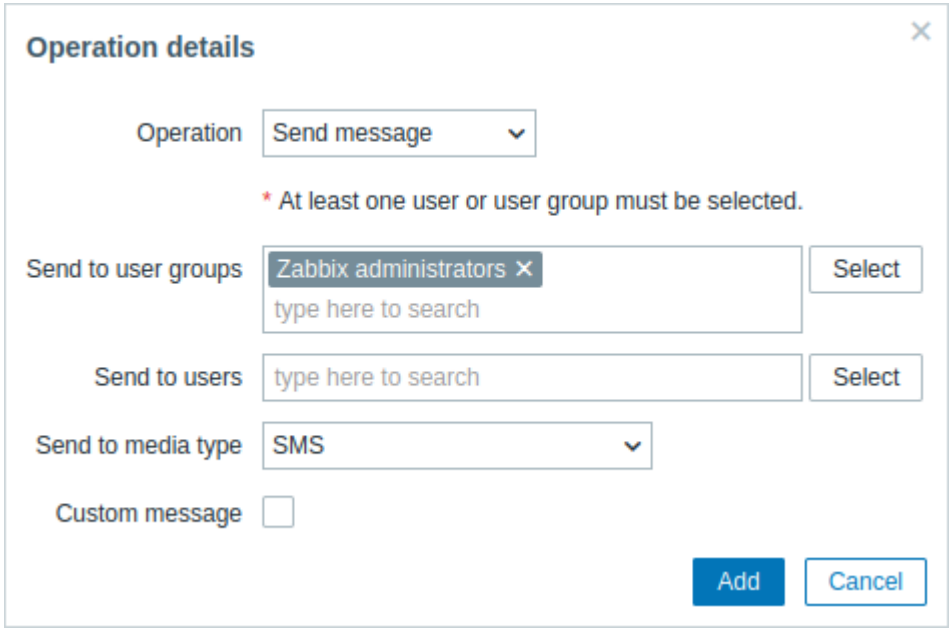
Настройка операции обновления

Для настройки операции обновления используйте следующие шаги:

1. Перейдите на вкладку *Операции (Operations)* в настройках действия.
2. Чтобы настроить новую операцию обновления, нажмите *Добавить (Add)* в блоке *Операции обновления (Update operations)*. Чтобы изменить существующую операцию, нажмите *Изменить (Edit)* сразу после операции.
3. Настройте подробности операции.



Детали операции обновления



Операции обновления предлагают тот же самый набор параметров, что и Операции восстановления.

Эскалации

Обзор

Используя эскалации, вы можете создавать пользовательские сценарии для отправки оповещений или выполнения удалённых команд.

С практической точки зрения эскалации означают, что:

- Пользователей можно информировать о новых проблемах незамедлительно.
- Оповещения могут повторяться до решения проблемы.
- Отправка оповещения может быть выполнена с задержкой.
- Оповещения могут эскалироваться другой «более высокой» группе пользователей.
- Удалённые команды могут быть выполнены незамедлительно или когда проблема не будет решена за длительный период времени.

Действия эскалируются на основании **шага эскалации**. Каждый шаг имеет длительность по времени.

Вы можете задать и длительность по умолчанию, и пользовательскую длительность для каждого отдельного шага. Минимальная длительность одного шага эскалации 60 секунд.

Вы можете начать действия, такие как отправка оповещения или выполнение команд, с любого шага. Шаг первый используется для немедленных действий. Если вы хотите отложить действие, вы можете назначить его на следующий шаг. Для каждого шага можно назначать несколько действий.

Количество шагов эскалаций не ограничено.

Эскалации задаются при настройке операции. Эскалации поддерживаются только для операций на проблемы, но не восстановления.

Прочие аспекты поведения эскалации

Рассмотрим, что происходит в различных обстоятельствах, если действие содержит несколько шагов эскалации.

Ситуация	Поведение
Указанный узел сети переходит в режим обслуживания после отправки первоначального уведомления о проблеме	В зависимости от настройки *Приостанавливать операции для подавленных проблем* в конфигурации действия все оставшиеся шаги эскалации выполняются либо с задержкой, вызванной периодом обслуживания, либо без задержки. Период обслуживания не отменяет операции.
*Период времени, заданный в условии действия **Период времени**, заканчивается после отправки первоначального уведомления*	Все оставшиеся шаги эскалации выполняются. Условие *Период времени* не может остановить операции; оно влияет на то, когда действия запускаются или не запускаются, но не на операции.
Проблема начинается во время обслуживания и продолжается (не устраняется) после окончания обслуживания	В зависимости от настройки *Приостанавливать операции для подавленных проблем* в конфигурации действия все шаги эскалации выполняются либо с момента окончания обслуживания, либо немедленно.
Проблема начинается во время обслуживания без данных и продолжается (не устраняется) после окончания обслуживания	Необходимо дождаться срабатывания триггера, прежде чем будут выполнены все шаги эскалации.
Различные эскалации следуют одна за другой с небольшим интервалом и перекрываются	Выполнение каждой новой эскалации заменяет предыдущую эскалацию, однако в предыдущей эскалации всегда выполняется как минимум один шаг эскалации. Такое поведение актуально для действий над событиями, создаваемыми при КАЖДОЙ оценке проблемы триггера.

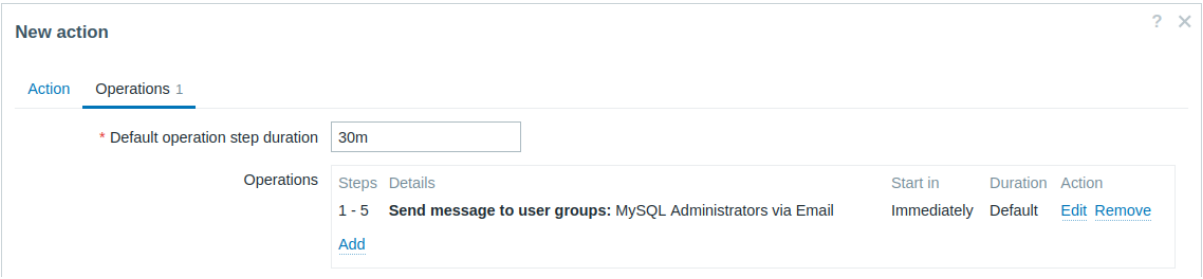
Ситуация	Поведение
Во время выполняющейся эскалации (например, при отправке сообщения), основанной на событии любого типа: - действие отключено На основе события триггера: - триггер отключен - узел сети или элемент данных отключен На основе внутреннего события о триггерах: - триггер отключен На основе внутреннего события об элементах данных или правилах обнаружения низкого уровня: - элемент данных отключен - узел сети отключен	Выполняющееся сообщение отправляется, после чего отправляется еще одно сообщение об эскалации. В начале текста последующего сообщения будет указано сообщение об отмене (*ПРИМЕЧАНИЕ: Эскалация отменена*) с указанием причины (например, *ПРИМЕЧАНИЕ: Эскалация отменена: действие «<Название действия>» отключено*). Таким образом получатель узнает, что эскалация отменена и дальнейшие шаги выполняться не будут. Это сообщение отправляется всем, кто ранее получил уведомления. Причина отмены также записывается в файл журнала сервера (начиная с уровня отладки 3=Warning). Обратите внимание, что сообщение *Эскалация отменена* также отправляется, если операции завершены, но настроены операции восстановления и они еще не выполнены.
Во время выполняющейся эскалации (например, при отправке сообщения) действие удаляется	Больше сообщения не отправляются. Информация записывается в файл журнала сервера (начиная с уровня отладки 3=Warning), например: `escalation canceled: action id:334 deleted`

Примеры эскалаций

Пример 1

Отправка повторяющихся оповещений каждые 30 минут (в общей сложности 5 раз) группе «MySQL администраторы». Для настройки:

- На вкладке *Операции (Operations)* задайте *Длительность шага операции по умолчанию (Default operation step duration)* значением «30m» (30 минут)
- Укажите шаги эскалаций *С (From)* «1» *До (To)* «5»
- Выберите группу «MySQL администраторы» получателями сообщений



Оповещения будут отправлены через 0:00, 0:30, 1:00, 1:30, 2:00 часов после начала проблемы (если, конечно, проблема не будет решена раньше).

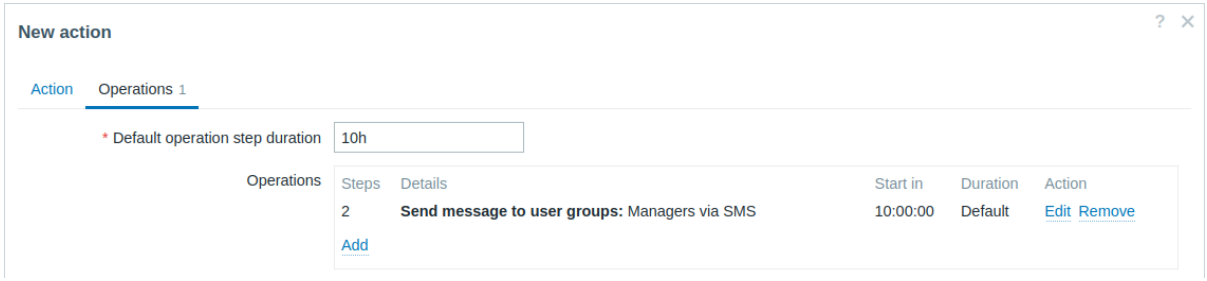
Если проблема решена и сообщение о восстановлении настроено, оно будет отправлено всем тем, кто получил хотя бы одно сообщение в этом сценарии эскалаций.

Note:
Если триггер, который вызвал активную эскалацию, был деактивирован, Zabbix отправит информационное сообщение об этом всем, кто уже получил оповещения.

Пример 2

Отправка оповещения с задержкой о давней проблеме. Для настройки:

- На вкладке *Операции (Operations)* задайте *Длительность шага операции по умолчанию (Default operation step duration)* значением «10h» (10 часов)
- Укажите шаги эскалации *С (From)* «2» *До (To)* «2»



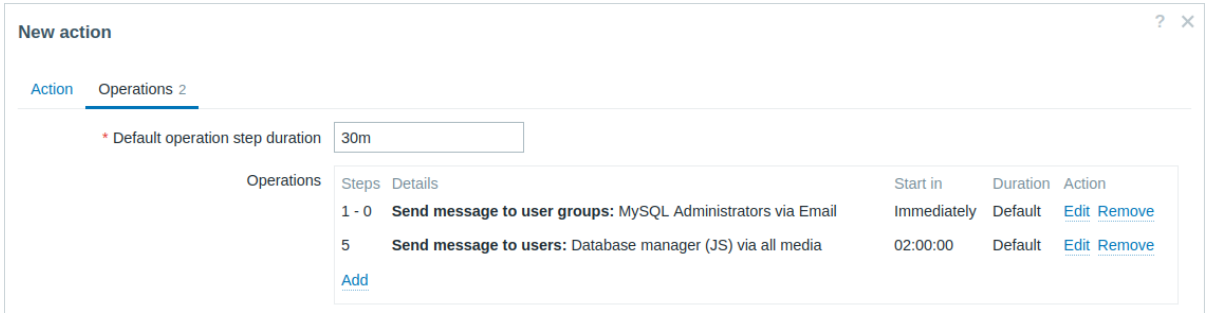
Оповещение будет отправлено только на Шаге 2 сценария эскалации, или через 10 часов после начала проблемы.

Вы можете изменить текст сообщения на что-то вроде «Проблема не решена уже более чем 10 часов».

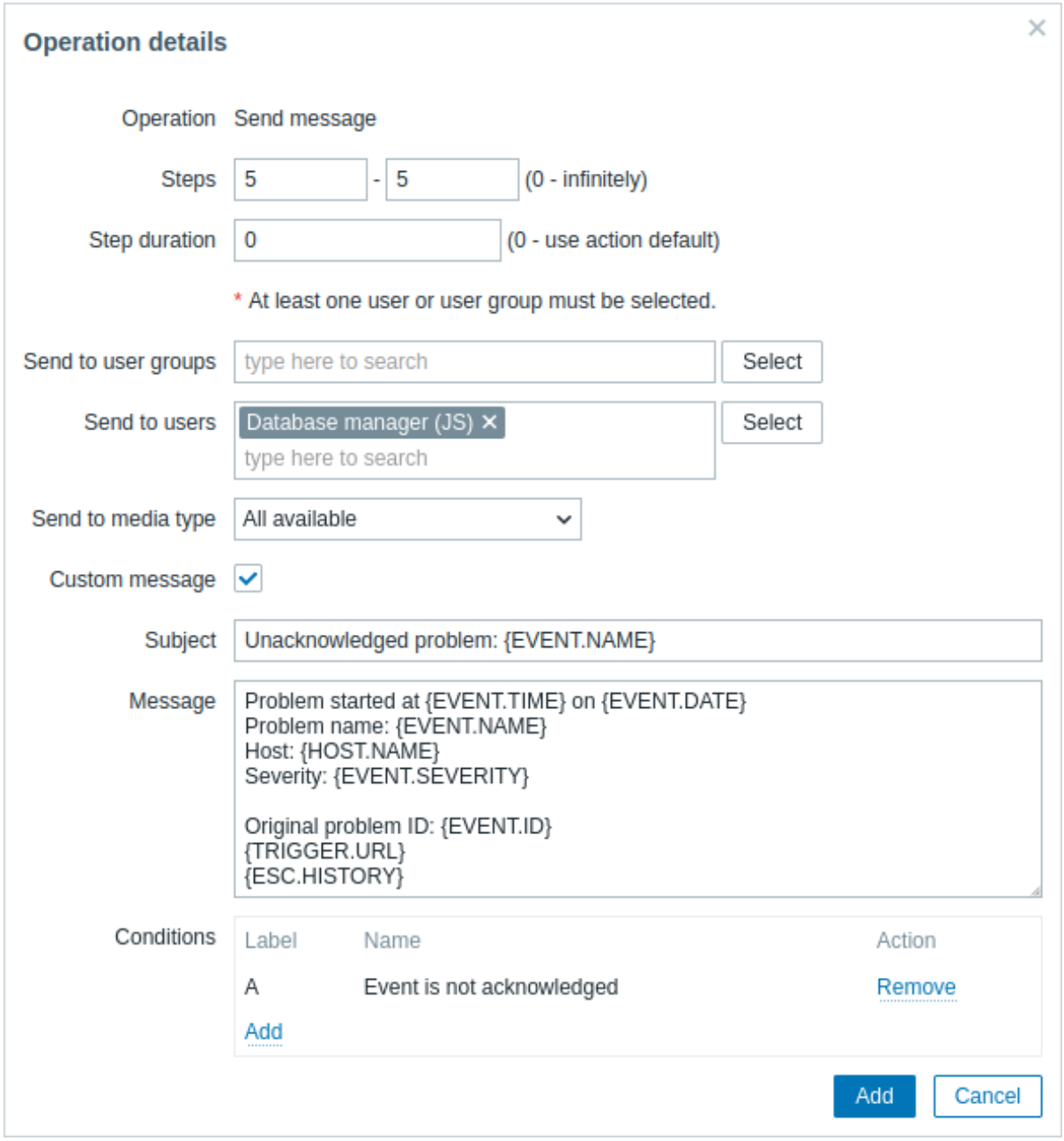
Пример 3

Эскалирование проблемы Руководству.

В первом примере мы настраивали периодическую отправку сообщений Администраторам MySQL. В данном же случае администраторы получат четыре сообщения до того, как проблема будет эскалирована Менеджеру баз данных. Обратите внимание, что менеджер получит сообщение только в случае, если проблема ещё не подтверждена, предположительно, никто не работает над ней.



Детали Операции 2:



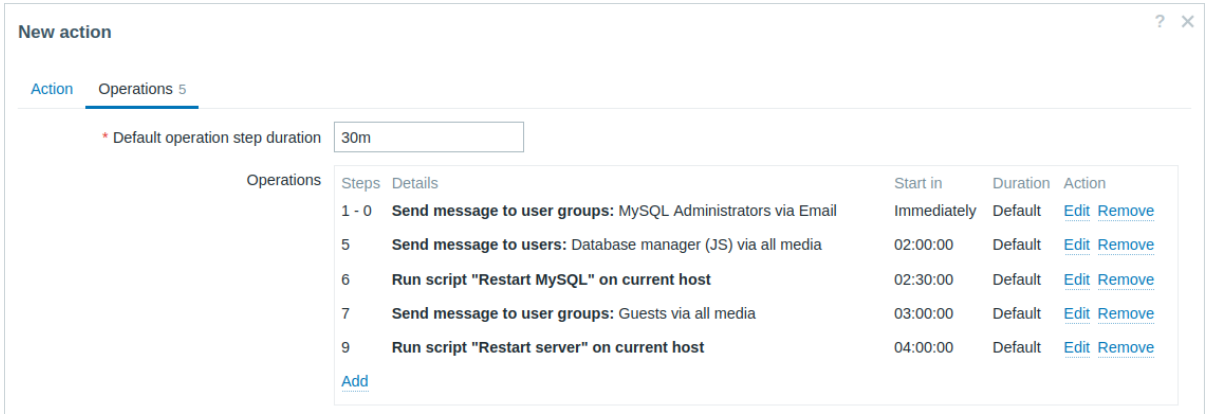
Обратите внимание на использование макроса {ESC.HISTORY} в сообщении. Этот макрос будет содержать информацию обо всех ранее выполненных шагах этой эскалации, таких как отправленные оповещения и выполненные команды.

Пример 4

Более сложный сценарий. После нескольких сообщений Администраторам MySQL и эскалации менеджеру, Zabbix попытается перезапустить базу данных MySQL. Это произойдёт, если через 2:30 часа проблема присутствует и не была подтверждена.

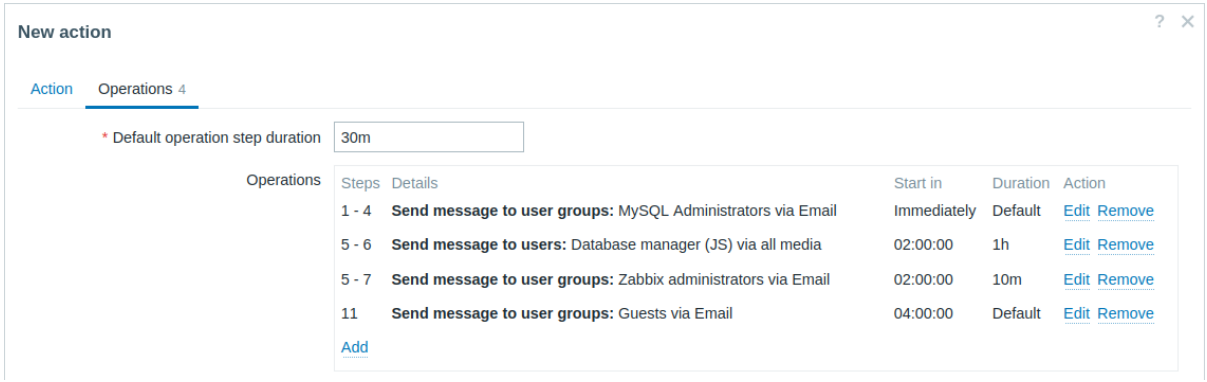
Если проблема всё ещё существует, спустя ещё 30 минут Zabbix отправит сообщение всем гостевым пользователям.

Если и это не поможет, спустя ещё час Zabbix перезагрузит сервер с этой базой данных MySQL (вторая удалённая команда), используя IPMI команды.



Пример 5

Эскалация с несколькими операциями, которые имеют перекрывающиеся диапазоны шагов и пользовательские интервалы. Длительность шага операции по умолчанию равна 30 минутам.



Оповещения будут отправлены в следующем порядке:

- Администраторам MySQL через 0:00, 0:30, 1:00, 1:30 после начала проблемы.
- Менеджеру баз данных через 2:00 и 2:10 (более короткая пользовательская длительность шага в 10 минут, заданная в последующей операции, перекрывает более длительную длительность шага в 1 час, настроенную здесь, как указано в разделе Детали операций для параметра *Длительность шага (Step duration)*, если шаги перекрываются).
- Zabbix администраторам через 2:00, 2:10 и 2:20 после начала проблемы (применяется пользовательская длительность шага, равная 10 минутам).
- Гостевым пользователям через 4:00 часа после начала проблемы (между шагами 8 и 11 имеет силу интервал по умолчанию, равный 30 минутам).

Получение уведомления о неподдерживаемых элементах данных

Обзор

В Zabbix можно получать уведомления о неподдерживаемых элементах данных.

Это часть концепции внутренних событий в Zabbix, которая позволяет пользователям получать уведомления в таких случаях. Внутренние события отражают изменение состояния:

- когда элементы данных переходят из состояния ’normal’ в ’unsupported’ (и обратно);
- когда триггеры переходят из состояния ’normal’ в ’unknown’ (и обратно);
- когда правила обнаружения низкого уровня переходят из состояния ’normal’ в ’unsupported’ (и обратно).

В этом разделе описано, как **получать уведомление**, когда элемент данных становится неподдерживаемым.

Настройка

В целом, процесс настройки оповещений должен быть знаком тем, кто раньше настраивал оповещения в Zabbix.

Шаг 1

Настройте какой-либо способ оповещения, такой как email, SMS или скрипт, для использования в оповещениях. Для выполнения этой задачи обратитесь к соответствующим разделам документации.

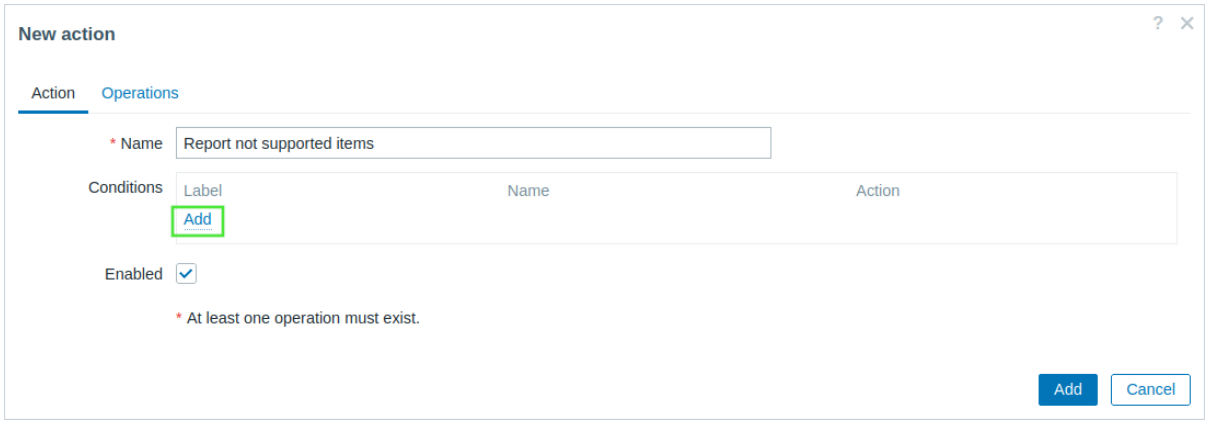
Attention:
При оповещении о внутренних событиях используется важность по умолчанию («Не классифицировано»), поэтому оставьте эту важность выбранной при настройке оповещения пользователя, если вы хотите получать оповещения о внутренних событиях.

Шаг 2

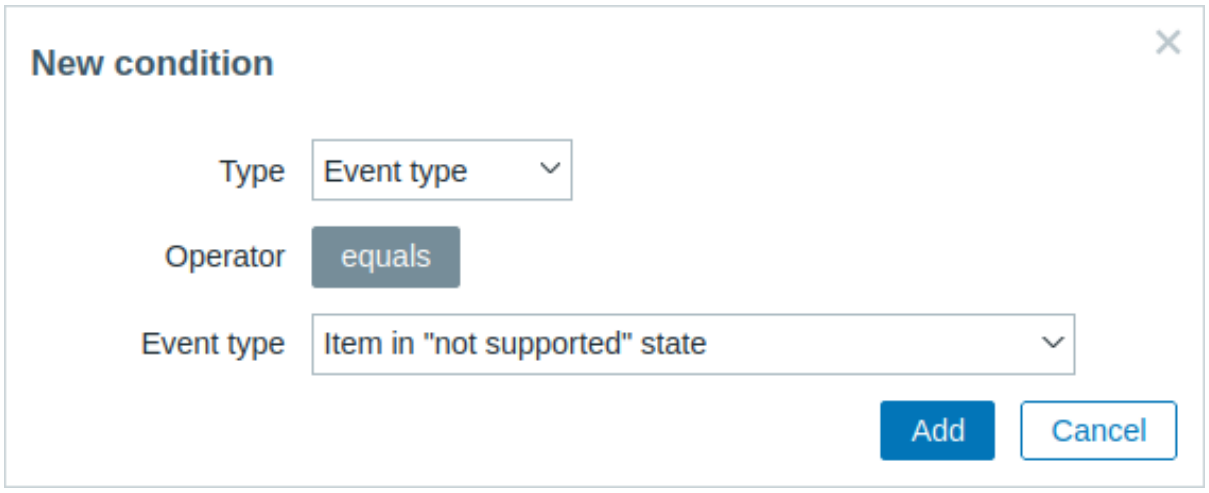
1. Перейдите в *Alerts* > *Actions* > *Internal actions*.
2. Нажмите *Create action* в правом верхнем углу страницы, чтобы открыть форму настройки действия.

Шаг 3

1. На вкладке *Действие* введите имя действия.
2. Нажмите *Добавить* в блоке *Условия*, чтобы добавить новое условие.



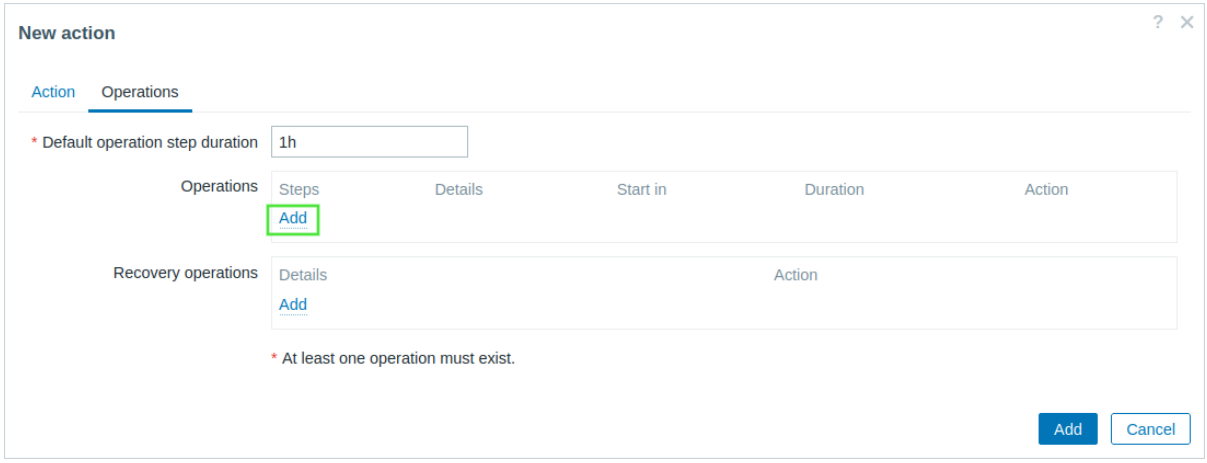
3. Во всплывающем окне *Новое условие* выберите «Тип события» в качестве типа условия, а затем выберите «Элемент данных в состоянии «не поддерживается»» в качестве типа события.



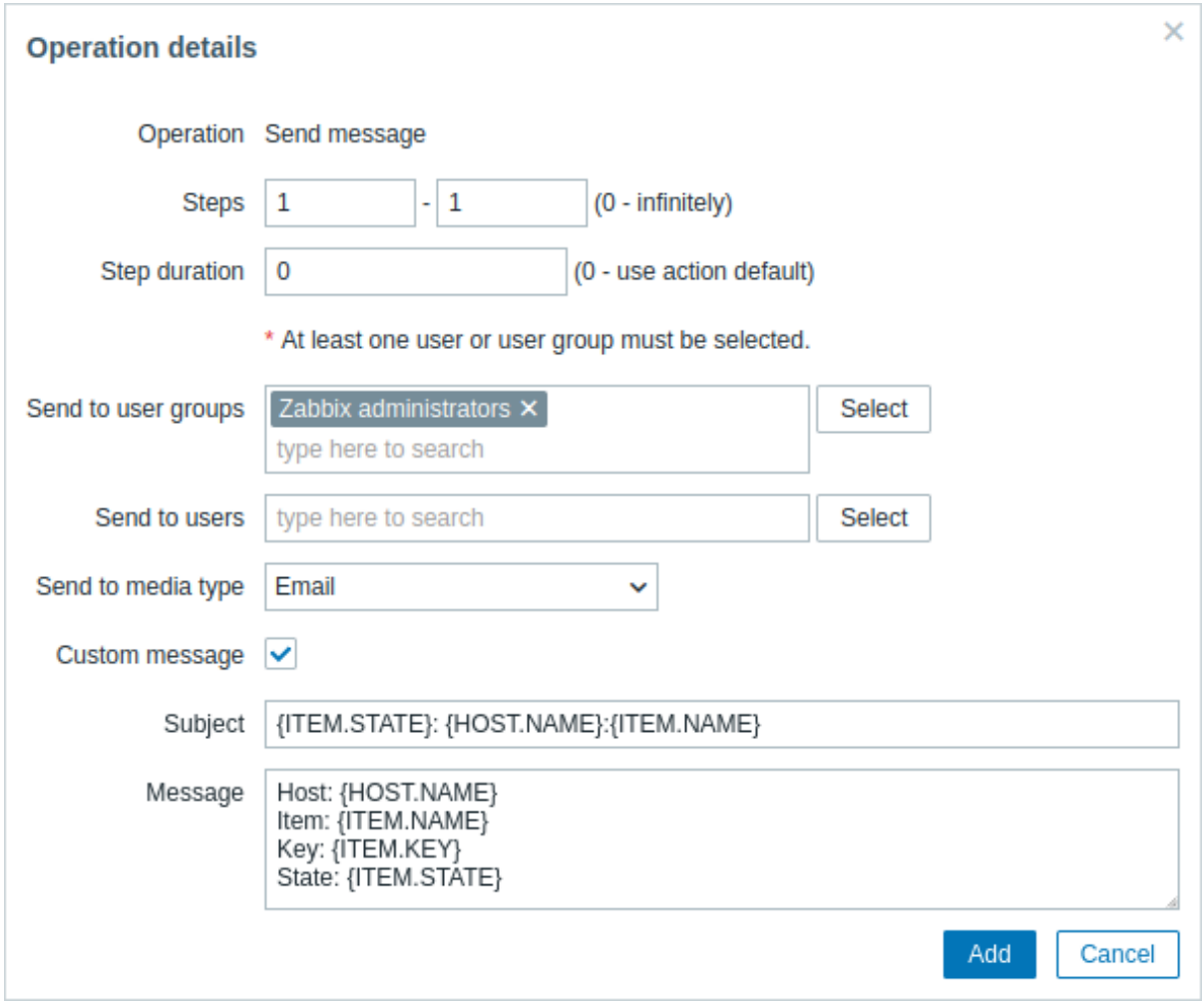
4. Нажмите *Добавить*, чтобы добавить условие в блок *Условия*.

Шаг 4

1. На вкладке *Операции* нажмите *Добавить* в блоке *Операции*, чтобы добавить новую операцию.



2. Выберите получателей сообщения (группы пользователей/пользователей) и тип носителя (или «Все»), которые будут использоваться для доставки.
3. Установите флажок *Пользовательское сообщение*, если хотите указать пользовательские тему и содержимое сообщения о проблеме.



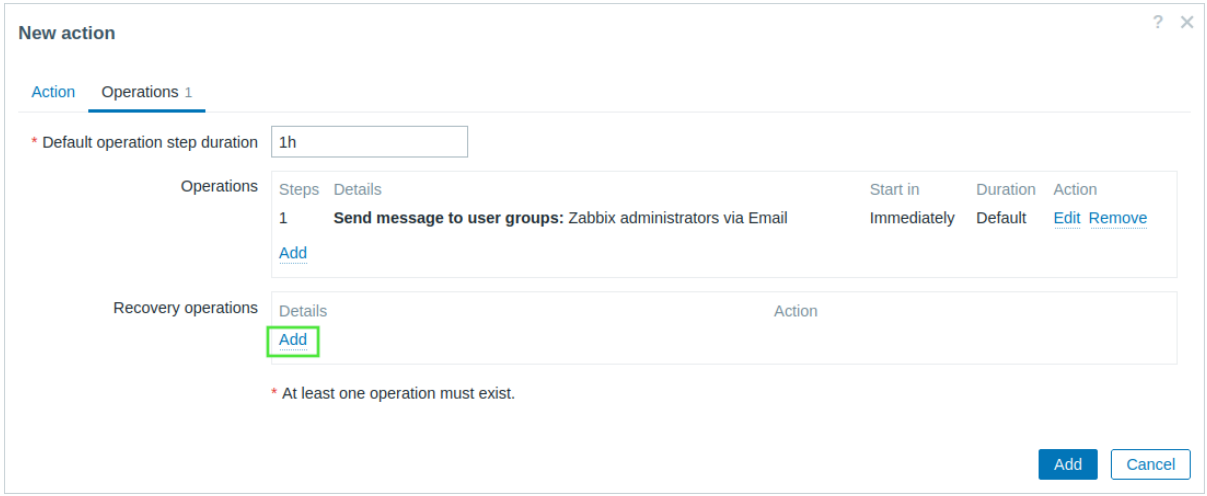
4. Нажмите *Добавить*, чтобы фактически добавить операцию в блок *Операции*.

Если вы хотите получать более одного уведомления, задайте длительность шага операции (интервал между отправляемыми сообщениями) и добавьте еще один шаг.

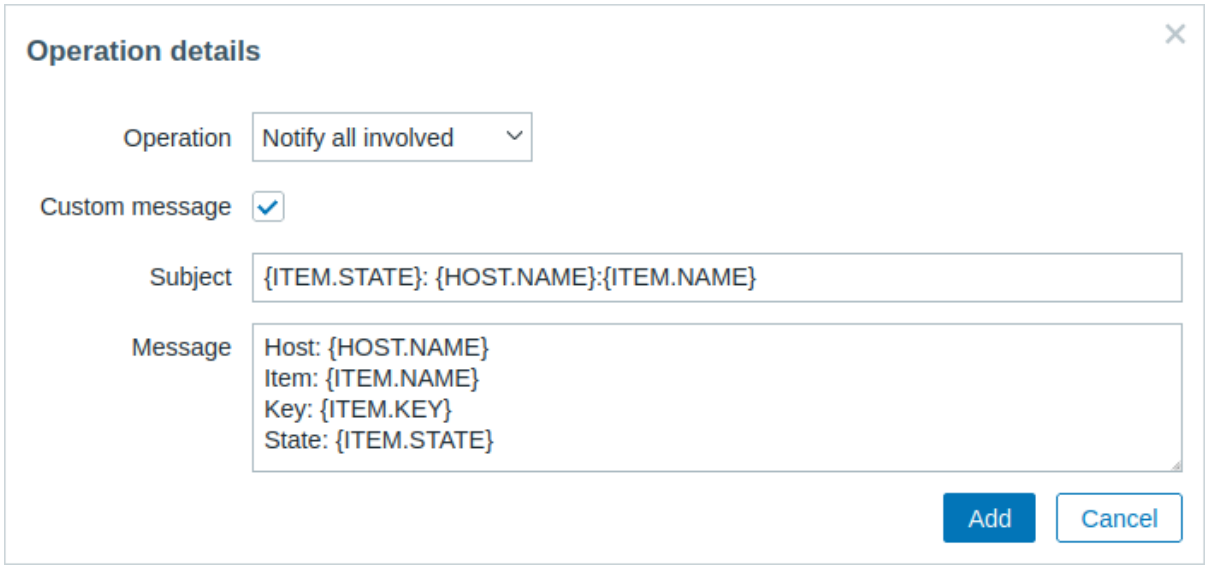
Шаг 5

Блок *Операции восстановления* позволяет настроить уведомление о восстановлении, когда элемент данных возвращается в нормальное состояние.

1. Нажмите *Добавить* в блоке *Операции восстановления*, чтобы добавить новую операцию восстановления.



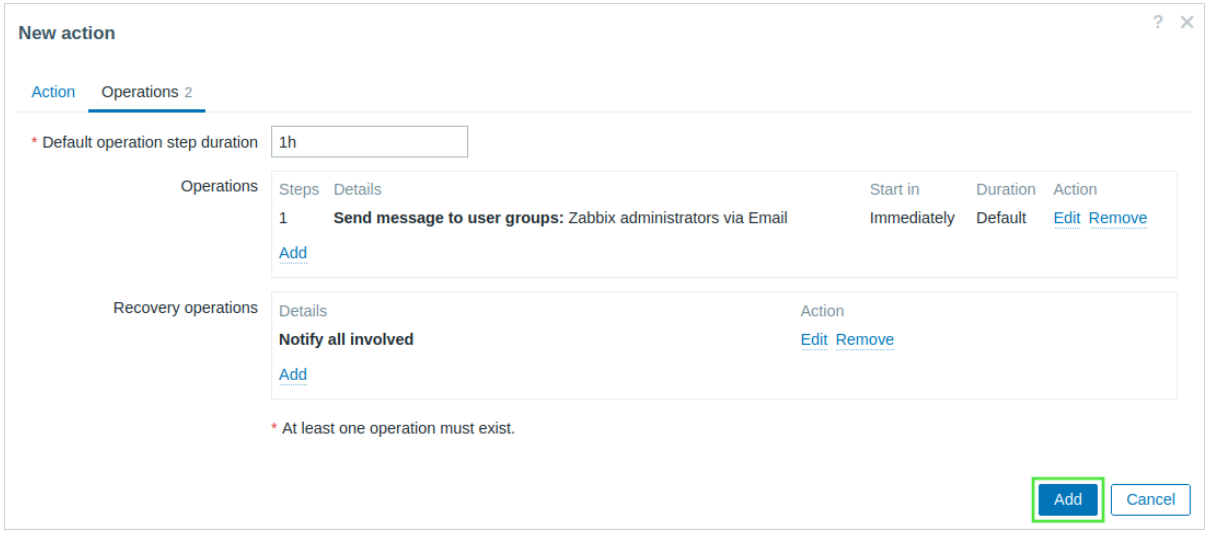
2. Выберите тип операции «Уведомить всех участников».
3. Установите флажок *Пользовательское сообщение*, если хотите указать пользовательские тему и содержимое сообщения о восстановлении.



4. Нажмите *Добавить* во всплывающем окне *Сведения об операции*, чтобы добавить операцию в блок *Операции восстановления*.

Шаг 6

После завершения нажмите кнопку *Add* в нижней части формы.



Вот и все, готово! Теперь вы можете ожидать получения первого уведомления от Zabbix, если какой-либо элемент данных перейдет в состояние неподдерживаемого.

Макросы

Обзор

Zabbix поддерживает ряд встроенных макросов, которые можно использовать в различных ситуациях. Эти макросы являются переменными, обозначаемыми специальным синтаксисом:

`{MACRO}`

Макросы преобразуются в определенное значение в зависимости от контекста.

Эффективное использование макросов позволяет экономить время и делает конфигурацию Zabbix более прозрачной.

В одном из типичных случаев макрос можно использовать в шаблоне. Например, триггер в шаблоне может называться «Слишком высокая загрузка процессора на {HOST.NAME}». Когда шаблон применяется к узлу сети, например к серверу Zabbix, при отображении триггера в разделе «Мониторинг» его название будет преобразовано в «Слишком высокая загрузка процессора на сервере Zabbix».

Макросы можно использовать в параметрах ключей элементов данных. Макрос можно использовать только в части параметра, например, `item.key[server_{HOST.HOST}_local]`. Заключать параметр в двойные кавычки не требуется, поскольку Zabbix самостоятельно обработает неоднозначные специальные символы, если они присутствуют в преобразованном макросе.

В Zabbix существуют и другие типы макросов.

Zabbix поддерживает следующие макросы:

- `{MACRO}` - встроенный макрос (см. полный список)
- `{<macro>.<func>(<params>)}` - функции макросов
- `{$MACRO}` - пользовательский макрос, при необходимости с контекстом
- `{#MACRO}` - макрос для обнаружения низкого уровня
- `{?EXPRESSION}` - макрос выражения

Функции макросов

Обзор

Функции макросов позволяют настраивать значения макросов (например, укорачивать их или извлекать определенные подстроки), что упрощает работу с ними.

Синтаксис функции макроса:

```
{macro.func(params)}
```

где

- **macro** - макрос, который нужно настроить;
- **func** - функция, которую нужно применить (см. поддерживаемые функции);
- **params** - список параметров функции, разделенных запятыми; они должны быть **заключены в двойные кавычки**, если:
  - начинаются с пробела или двойной кавычки;
  - содержат закрывающую скобку или запятую.

Например:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

Функции макросов поддерживаются для

- встроенных макросов
- пользовательских макросов
- макросов низкоуровневого обнаружения
- макросов выражений

Функции макросов можно использовать во всех местах, где поддерживаются перечисленные макросы. Это относится ко всем случаям, кроме тех, где явно указано, что ожидается только макрос (например, при настройке макросов узла сети или фильтров правила низкоуровневого обнаружения).

Для каждого макроса поддерживается только одна функция; цепочки из нескольких функций макросов не поддерживаются.

Note:
См. примеры экранирования для случаев, когда функции макросов используются внутри других контекстов (функции, ключа элемента данных, другого макроса и т. д.).

Поддерживаемые функции

Функции перечислены без дополнительной информации.
Нажмите функцию, чтобы просмотреть подробные сведения.

Функция	Описание
btoa	Кодирование значения макроса в формат Base64.
fmtnum	Форматирование числа для управления количеством знаков после десятичной точки.
fmttime	Форматирование времени.
htmldecode	Декодирование значения макроса из HTML-кодировки.
htmlencode	Кодирование значения макроса в HTML-кодировку.
iregsub	Извлечение подстроки по совпадению с регулярным выражением (без учета регистра).
jsonpath	Возврат результата JSONPath.
lowercase	Преобразование символов значения макроса в нижний регистр.
regrepl	Замена символа или подстроки в значении макроса.
regsub	Извлечение подстроки по совпадению с регулярным выражением (с учетом регистра).
tr	Транслитерация символов значения макроса.
uppercase	Преобразование символов значения макроса в верхний регистр.
urldecode	Декодирование значения макроса из URL-кодировки.
urlencode	Кодирование значения макроса в URL-кодировку.
xmlxpath	Возврат результата XML XPath.

Подробности функций

Необязательные параметры функций обозначены угловыми скобками (< >).

btoa

Кодирование значения макроса в кодировку Base64. Кодировка Base64 — это метод представления двоичных данных в виде текста, полезный для хранения и безопасной передачи двоичного содержимого по текстовым протоколам.

Пример:

```
{{ITEM.VALUE}.btoa()} -         Base64      ,       "zabbix",   "emFiYml4"
```

fmtnum(цифры)

Форматирование числа для управления количеством цифр, выводимых после десятичной точки.

Параметры:

- **цифры** — количество цифр после десятичной точки. Допустимый диапазон: 0-20. Конечные нули будут обрезаны.

Примеры:

```
{{ITEM.VALUE}.fmtnum(2)} —      «24.35»              «24.3483523»
{{ITEM.VALUE}.fmtnum(0)} —      «24»                 «24.3483523»
```

fmttime(format,<time_shift>)

Форматирование времени.
 Обратите внимание, что эту функцию можно использовать с макросами, которые преобразуются в значение в одном из следующих форматов времени:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (стандарт ISO8601)
- UNIX timestamp

Параметры:

- **format** - обязательная строка формата, совместимая с форматированием функции `strftime`;

- **time_shift** (необязательно) - сдвиг времени, применяемый ко времени перед форматированием; должен начинаться с -<N><time_unit> или +<N><time_unit>, где:
  - N - количество единиц времени, которые нужно прибавить или вычесть;
  - time_unit - h (час), d (день), w (неделя), M (месяц) или y (год).

Комментарии:

- Параметр time_shift поддерживает многошаговые операции со временем и может включать /<time_unit> для сдвига к началу единицы времени (/d - полночь, /w - первый день недели (понедельник), /M - первый день месяца и т. д.). Примеры: -1w - ровно 7 дней назад; -1w/w - понедельник предыдущей недели; -1w/w+1d - вторник предыдущей недели.
- Операции со временем вычисляются слева направо без приоритетов. Например, -1M/d+1h/w будет разобрано как ((-1M/d)+1h)/w.

Примеры:

```
{{TIMESTAMP}.fmttime(%B)} -      "October"                "1633098961"
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} -      "1 September"                "1633098961"
{{EVENT.TIMESTAMP}.fmttime("%x %X")} -                                ;      , "07/15/26 10:25:39"
{{EVENT.TIME}.fmttime("%x %X")} -                                   ;      , "07/15/26 10:25:39"            "17
```

htmldecode

Декодирование значения макроса из HTML-кодировки.

Поддерживаются следующие символы:

Значение	Декодированное значение
&	&
<	<
>	>
"	"
'	'
'	'

Пример:

```
{{ITEM.VALUE}.htmldecode()} -       HTML-              ,              "&lt;"   "<"
```

htmlencode

Кодирование значения макроса в HTML-кодировку.

Поддерживаются следующие символы:

Значение	Закодированное значение
&	&
<	<
>	>
"	"
'	'

Пример:

```
{{ITEM.VALUE}.htmlencode()} -          HTML         ,    "<",   "&lt;"
```

iregsub(шаблон,вывод)

Извлечение подстроки помощью сопоставления с регулярным выражением (без учёта регистра).

Parameters:

- **шаблон** — регулярное выражение для поиска совпадения;
- **вывод** — опции вывода. Для захвата групп поддерживаются местозаменители **\1 - \9**. Заменитель **\0** вернёт совпавший текст.

Комментарии:

- Если нет совпадений с регулярным выражением, функция возвращает пустую строку.

- Если шаблон функции не является корректным регулярным выражением, то макрос раскрывается в «НЕИЗВЕСТНО» (за исключением макросов низкоуровневого обнаружения, где в этом случае функция будет проигнорирована и макрос останется нераскрытым).
- Ссылки на несуществующие скобочные группы в строке замены заменяются пустой строкой.

Пример:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} —           «ERROR»,               «fail», «er
```

jsonpath(pattern,<default>)

Возвращает результат JSONPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **pattern** — обязательный шаблон;
- **default** — необязательное резервное значение, если запрос JSONPath не возвращает данных. Обратите внимание, что при возникновении других ошибок возвращается ошибка.

Комментарии:

- Функция завершается с ошибкой, если:
  - JSON имеет недопустимый формат.
  - Запрос JSONPath завершается с ошибкой.
  - Результат JSONPath не содержит данных, а параметр `default` не указан.

Пример:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

lowercase

Преобразование всех символов значения макроса в нижний регистр.
Работает с однобайтовыми наборами символов (такими как ASCII) и не поддерживает UTF-8.

Пример:

```
{{ITEM.VALUE}.lowercase()} -            ,      "Zabbix SERVER",   "zabbix server" (        )
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Замена символа/подстроки в значении макроса.

Параметры:

- **pattern** - регулярное выражение для сопоставления;
- **replacement** - строка замены. В строках замены поддерживаются заполнители **\1 - \9** для захвата групп.

Комментарии:

- Все совпадения, соответствующие шаблону, заменяются.
- Шаблоны и замены обрабатываются последовательно, при этом каждая последующая пара применяется в соответствии с результатом предыдущей замены.
- Ссылки на несуществующие группы захвата в строке замены заменяются пустой строкой.

Примеры:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} -      "oldParam"   "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\1")} -
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} -                        (     ,          "C:\"   "C:")
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} -
```

regsub(шаблон,вывод)

Извлечение подстроки помощью сопоставления с регулярным выражением (с учётом регистра).

Параметры:

- **шаблон** — регулярное выражение для поиска совпадения;
- **вывод** — опции вывода. Для захвата групп поддерживаются местозаменители **\1 - \9**. Заменитель **\0** вернёт совпавший текст.

Комментарии:

- Если нет совпадений с регулярным выражением, функция ничего не возвращает.

- Если шаблон функции не является корректным регулярным выражением, то макрос раскрывается в «НЕИЗВЕСТНО» (за исключением макросов низкоуровневого обнаружения, где в этом случае функция будет проигнорирована и макрос останется нераскрытым).
- Ссылки на несуществующие скобочные группы в строке замены заменяются пустой строкой.

Примеры:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} —           «Problem ID: 123»,                  «12
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} —           «ERROR»,               «fail», «er
```

Смотрите ещё примеры.

tr(characters,replacement)

Транслитерация символов значения макроса.

- **characters** — набор символов для замены;
- **replacement** — набор позиционно соответствующих символов замены.

Примеры:

```
{{ITEM.VALUE}.tr(abc, xyz)} -              "a"   "x", "b"   "y", "c"   "z"
{{ITEM.VALUE}.tr(abc, xyzq)} -              "a"   "x", "b"   "y", "c"   "z" ("q"       )
{{ITEM.VALUE}.tr(abcde, xyz)} -              "a"   "x", "b"   "y", "c"   "z", "d"   "z", "e"   "z" (
{{ITEM.VALUE}.tr("\\\'", "\/\"")} -                                       ,
{{ITEM.VALUE}.tr(A-Z,a-z)} -
{{ITEM.VALUE}.tr(0-9a-z,*)} -                         "*"
{{ITEM.VALUE}.tr(0-9,ab)} -              0   "a",           1, 2, 3, 4, 5, 6, 7, 8   9   "b"
{{ITEM.VALUE}.tr(0-9abcA-L,*)} -           ,      "abc"        A-L   "*"
{{ITEM.VALUE}.tr("\n","*")} -                        *
{{ITEM.VALUE}.tr("e", "\n")} -        "e"
```

Чтобы включить буквальные символы:

```
            -                  \\
          -                  \'
         -                  \"
```

Поддерживаемые escape-последовательности с обратной косой чертой:

```
\\\\ => \\ -
\\a  => \a - alert
\\b  => \b - backspace
\\f  => \f -
\\n  => \n -
\\r  => \r -
\\t  => \t -
\\v  => \v -
```

uppercase

Преобразование всех символов значения макроса в верхний регистр.
Работает с однобайтовыми наборами символов (такими как ASCII) и не поддерживает UTF-8.

Пример:

```
{{ITEM.VALUE}.uppercase()} -            ,      "Zabbix Server",   "ZABBIX SERVER" (        )
```

urldecode

Декодирование значения макроса из URL-кодировки.

Пример:

```
{{ITEM.VALUE}.urldecode()} -      URL-            ,      "%2F"   "/"
```

urlencode

Кодирование значения макроса в URL-кодировку.

Пример:

```
{{ITEM.VALUE}.urlencode()} -      URL-               ,    "/",   "%2F"
```

xmlxpath(pattern,<default>)

Возвращает результат XML XPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **pattern** — обязательный шаблон;
- **default** — необязательное резервное значение, если запрос XPath не возвращает данных. Обратите внимание, что при возникновении других ошибок возвращается состояние сбоя.

Комментарии:

- Функция завершается с ошибкой, если:
  - XML имеет недопустимый формат.
  - Запрос XPath завершается с ошибкой.
  - результат XPath не содержит данных, а параметр `default` не указан.
  - Zabbix скомпилирован без поддержки XML.

Пример:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

Дополнительные примеры

В таблице ниже показано больше примеров использования функций макросов.

`{#IFALIAS}` — это LLD-макрос и определяется только в контекстах низкоуровневого обнаружения (правила обнаружения, прототипы и созданные на их основе элементы данных/триггеры). При использовании вне LLD токен останется неразвернутым.

Функция макроса	Полученное значение	Вывод
`{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}`	`123Log line`	`Problem`
`{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}`	`123 Log line`	`Problem`
`{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}`	`Log line`	`Problem ID:`
`{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}`	`MySQL crashed errno 123`	`Problem ID: MySQL_123`
`{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}`	`123 Log line`	`*UNKNOWN*` (некорректное регулярное выражение)
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}`	`customername_1`	`customername`
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}`	`customername_1`	`1`
`{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}`	`customername_1`	`{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}` (некорректное регулярное выражение)
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	`customername_1`	`{$MACRO:"customername"}`
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}`	`customername_1`	`{$MACRO:"1"}`
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	`customername_1`	`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}` (некорректное регулярное выражение)
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"`	`customername_1`	`"{$MACRO:"\customername\"}"`
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \2)}\"}"`	`customername_1`	`"{$MACRO:"\1\"}"`
`"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+", \1)}\"}"`	`customername_1`	`"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"` (некорректное регулярное выражение)

Просмотр полных значений элементов данных

Длинные значения развернутых макросов {ITEM.VALUE} и {ITEM.LASTVALUE} для элементов данных типа text/log в некоторых разделах веб-интерфейса обрезаются до 20 символов. Чтобы увидеть полные значения этих макросов, можно использовать функции макросов, например:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

См. также: {ITEM.VALUE} и {ITEM.LASTVALUE} подробности о макросах.

Пользовательские макросы

Обзор

Пользовательские макросы поддерживаются в Zabbix для большей гибкости в дополнение к макросам, поддерживаемым из коробки.

Пользовательские макросы можно определять на глобальном уровне, уровне шаблона и уровне узла сети. Эти макросы имеют специальный синтаксис:

`{$MACRO}`

Zabbix разрешает макросы в соответствии со следующими приоритетами:

1. Макросы на уровне узла сети (проверяются в первую очередь).
2. Макросы, определенные для шаблонов первого уровня узла сети (то есть шаблонов, непосредственно связанных с узлом сети), отсортированные по ID шаблона.
3. Макросы, определенные для шаблонов второго уровня узла сети, отсортированные по ID шаблона.
4. Макросы, определенные для шаблонов третьего уровня узла сети, отсортированные по ID шаблона, и так далее.
5. Глобальные макросы (проверяются в последнюю очередь).

Иными словами, если макрос не существует для узла сети, Zabbix попытается найти его в шаблонах узла сети, увеличивая глубину вложенности. Если макрос по-прежнему не найден, будет использован глобальный макрос, если он существует.

Warning:
Если макрос с **одинаковым именем** существует в нескольких связанных шаблонах одного уровня, будет использован макрос из шаблона с наименьшим ID. Поэтому наличие макросов с одинаковыми именами в нескольких шаблонах представляет собой риск конфигурации.

Если Zabbix не может найти макрос, макрос не будет разрешен.

Attention:
Макросы (включая пользовательские макросы) намеренно остаются неразрешенными в разделе «Конфигурация» (например, в списке триггеров), чтобы сделать сложную конфигурацию более прозрачной.

Пользовательские макросы можно использовать в следующих местах:

- имени элемента данных
- параметре ключа элемента данных
- интервалах обновления элемента данных и гибких интервалах
- имени и описании триггера
- параметрах и константах выражения триггера (см. примеры)
- многих других местах — см. полный список

Общие случаи использования глобальных макросов и макросов уровня узлов сети

- используйте глобальные макросы в нескольких местах; затем измените значения этих макросов, и изменения настроек применятся во всех местах за «одно нажатие»
- получение преимуществ от шаблонов с уникальными для узлов сети атрибутами: пароли, номера портов, имена файлов, регулярные выражения и т.д.

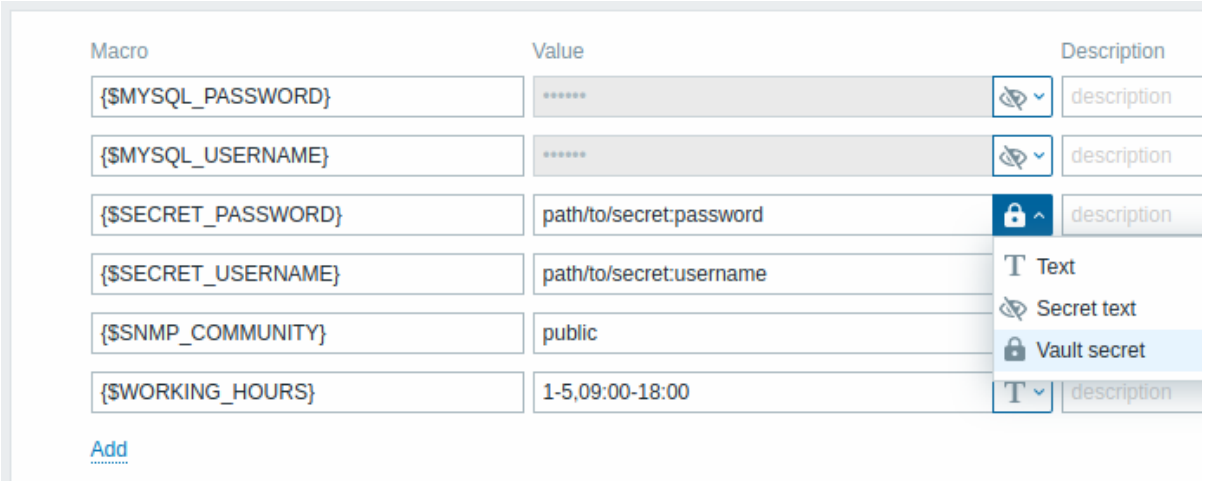
Note:
Рекомендуется использовать макросы узла сети вместо глобальных макросов, поскольку добавление, обновление или удаление глобальных макросов приводит к инкрементальному обновлению конфигурации для всех узлов сети. Дополнительную информацию см. в разделе Пассивные и активные проверки агента.

Конфигурация

Чтобы определить пользовательские макросы, перейдите в соответствующее место на веб-интерфейсе:

- Для глобальных макросов откройте *Администрирование > Макросы*.
- Для макросов уровня узла сети и шаблона откройте свойства узла сети или шаблона и найдите вкладку *Макросы*.

Пользовательский макрос имеет следующие атрибуты:



Parameter	Description
Macro	Имя макроса. Имя должно быть заключено в фигурные скобки и начинаться со знака доллара. Пример: {$FRONTEND_URL}. В именах макросов допускаются следующие символы: **A-Z** (только заглавные) , **0-9** , _ , **.**
Value	Значение макроса. Поддерживаются три типа значений: **Text** (по умолчанию) - обычное текстовое значение **Secret text** - значение скрывается звездочками **Vault secret** - значение содержит путь/запрос к vault secret. Чтобы изменить тип значения, нажмите кнопку в конце поля ввода значения. Максимальная длина значения пользовательского макроса — 2048 символов.
Description	Текстовое поле, используемое для предоставления дополнительной информации об этом макросе.

При настройке пользовательских макросов **обратите внимание** на следующие особенности, зависящие от контекста:

- если пользовательские макросы используются в элементах данных или триггерах шаблона, рекомендуется добавить эти макросы также в шаблон (даже если они определены глобально); в этом случае макросы типа *Text* будут работать ожидаемым образом после экспорта шаблона в XML и импорта в другую систему (значения секретных макросов не экспортируются)
- если пользовательские макросы используются в выражениях триггеров, такие макросы будут подставляться только при ссылке на параметр или константу; они НЕ будут подставляться при ссылке на узел сети, ключ элемента данных, функцию, оператор или другое выражение триггера (секретные макросы нельзя использовать в выражениях триггеров)
- если пользовательские макросы используются на узле сети, для которого есть правило низкоуровневого обнаружения с прототипами узлов сети, обнаруженные узлы сети унаследуют все пользовательские макросы, определенные на этом узле сети

Примеры

Пример 1

Использование макроса уровня узла сети в ключе элемента данных «Состояние демона SSH»:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Этот элемент данных можно назначить нескольким узлам сети при условии, что значение {$SSH_PORT} определено на этих узлах.

Пример 2

Использование макроса уровня узла сети в триггере «Слишком высокая загрузка ЦП»:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Такой триггер создается в шаблоне, а не редактируется на отдельных узлах сети.

Note:
Если вы хотите использовать количество значений в качестве параметра функции (например, `max(/host/key,#3)`), включите знак решетки в определение макроса следующим образом: `SOME_PERIOD => #3`.

Пример 3

Использование двух макросов в триггере «Слишком высокая загрузка ЦП»:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Обратите внимание, что макрос можно использовать в качестве параметра функции триггера, в данном примере функции `min()`.

Пример 4

Синхронизируйте условие недоступности агента с интервалом обновления элемента данных:

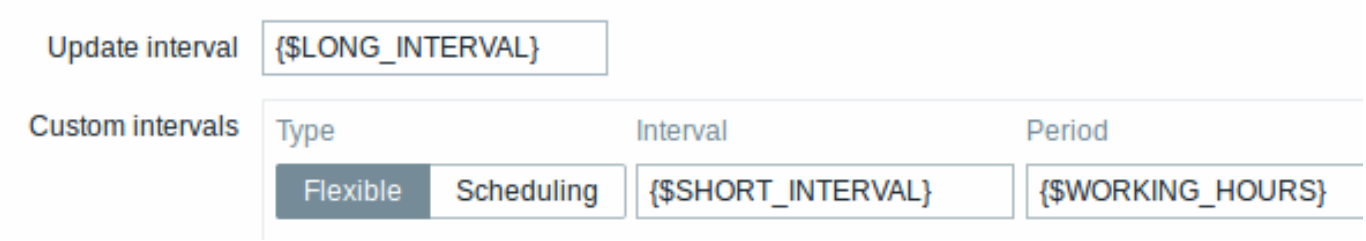
- определите макрос `{$INTERVAL}` и используйте его в интервале обновления элемента данных;
- используйте `{$INTERVAL}` в качестве параметра триггера недоступности агента:

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

Пример 5

Централизуйте настройку рабочих часов:

- создайте глобальный макрос `{$WORKING_HOURS}` со значением `1-5,09:00-18:00`;
- используйте его в поле *Рабочее время* в разделе *Администрирование* > *Общие* > *Веб-интерфейс*;
- используйте его в поле *Когда активен* в разделе *Пользователи* > *Пользователи*, на вкладке *Мультимедиа* пользователя;
- используйте его для настройки более частого опроса элементов данных в рабочие часы:

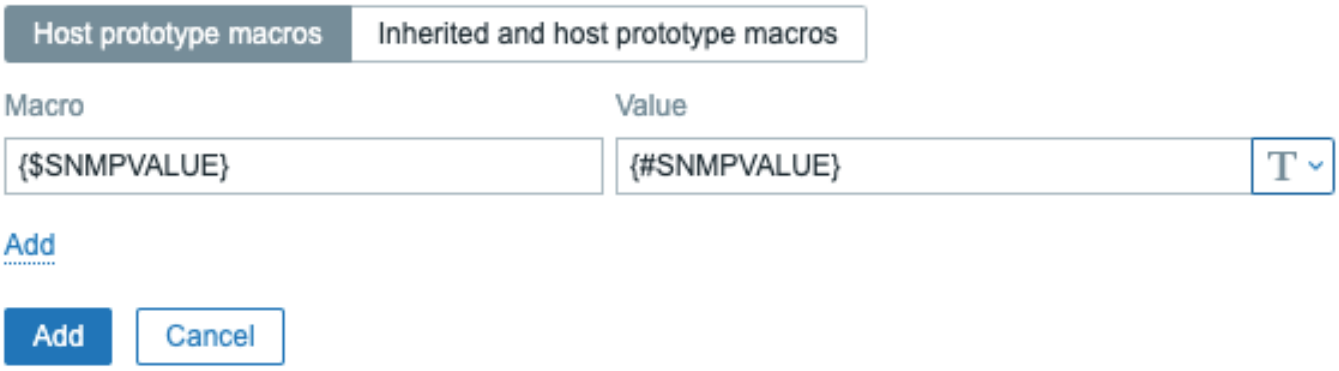


- используйте его в условии действия *Период времени*;
- при необходимости скорректируйте рабочее время в разделе *Администрирование* > *Макросы*.

Пример 6

Используйте макрос прототипа узла сети для настройки элементов данных обнаруженных узлов сети:

1. В прототипе узла сети определите пользовательский макрос `{$SNMPVALUE}`, указав в качестве значения макрос обнаружения низкого уровня `{#SNMPVALUE}`:



2. Назначьте шаблон *Generic SNMPv2* прототипу узла сети.
3. Используйте `{$SNMPVALUE}` в поле *SNMP OID* элементов данных шаблона *Generic SNMPv2*.

Контекст пользовательского макроса

См. пользовательские макросы с контекстом.

Пользовательские макросы с контекстом

Обзор

В пользовательских макросах можно использовать необязательный контекст, позволяющий переопределить значение по умолчанию значением, заданным для конкретного контекста.

Контекст добавляется к имени макроса; синтаксис зависит от того, является ли контекст статическим текстовым значением:

```
{$MACRO:"static text"}
```

или регулярным выражением:

```
{$MACRO:regex:"regular expression"}
```

Обратите внимание, что макрос с контекстом в виде регулярного выражения можно определить только в конфигурации пользовательских макросов. Если префикс `regex:` используется в другом месте в качестве контекста пользовательского макроса, например в выражении триггера, он будет интерпретирован как статический контекст.

Заключать контекст в кавычки необязательно (см. также важные замечания).

Примеры контекста макросов:

Пример	Описание
`{$LOW_SPACE_LIMIT}`	Пользовательский макрос без контекста.
`{$LOW_SPACE_LIMIT:/tmp}`	Пользовательский макрос с контекстом (статическая строка).
`{$LOW_SPACE_LIMIT:regex:"^/tmp$"}`	Пользовательский макрос с контекстом (регулярное выражение). То же, что и `{$LOW_SPACE_LIMIT:/tmp}`.
`{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}`	Пользовательский макрос с контекстом (регулярное выражение). Соответствует всем строкам, начинающимся с /var/log/.

Варианты использования

Пользовательские макросы с контекстом можно определить, чтобы реализовать более гибкие пороги в выражениях триггеров (на основе значений, полученных с помощью обнаружения на низком уровне). Например, можно определить следующие макросы:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

Затем макрос обнаружения на низком уровне может использоваться как контекст макроса в прототипе триггера для обнаружения смонтированных файловых систем:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

После обнаружения в триггерах будут применяться разные пороги по низкому объему свободного места в зависимости от обнаруженных точек монтирования или типов файловых систем. События о проблеме будут сгенерированы, если:

- в папке /home свободно менее 20% дискового пространства
- в папках, соответствующих шаблону regexp (например, /etc, /tmp или /var), свободно менее 30% дискового пространства
- в папках, не соответствующих шаблону regexp и не являющихся /home, свободно менее 10% дискового пространства

Важные замечания

- Если существует более одного пользовательского макроса с контекстом, Zabbix сначала попытается сопоставить макросы с простым контекстом и только затем макросы с контекстом в виде регулярного выражения в неопределённом порядке.

Warning:
Не создавайте разные макросы с контекстом, которые соответствуют одной строке, чтобы избежать неопределённого поведения.

- Если макрос со своим контекстом не найден на узле сети, присоединённых шаблонах или на глобальном уровне, тогда будет произведён поиск макроса без контекста.
- В контекстах макроса поддерживаются только макросы низкоуровневого обнаружения. Любые другие макросы игнорируются и обрабатываются как текстовые значения.

Технически, контекст макроса задаётся с использованием правил, похожих на параметры ключей элементов данных, за исключением того, что контекст макроса при наличии символа , не обрабатывается как несколько параметров:

- Контекст макроса необходимо заключать в кавычки ", если контекст содержит символ } или он начинается с символа ". Кавычки внутри заключённого в кавычки контекста необходимо экранировать при помощи символа \.
- Сам символ \ не экранируется, что означает, что невозможно задать заключённый в кавычки контекст, оканчивающийся на символ \ — макрос {$MACRO:"a:\b\c\"} ошибочный.
- Пробелы в начале контекста игнорируются, пробелы в конце не игнорируются:
  - Например, {$MACRO:A} тоже самое, что и {$MACRO: A}, но не то же самое, что {$MACRO:A }.
- Все пробелы до открывающих кавычек и после закрывающих кавычек игнорируются, но все пробелы внутри кавычек не игнорируются:
  - Макросы {$MACRO:"A"}, {$MACRO: "A"}, {$MACRO:"A" } и {$MACRO: "A" } одинаковы, но макросы {$MACRO:"A"} и {$MACRO:" A "} не одинаковы.

Следующие макросы идентичны, так как имеют один и тот же контекст: {$MACRO:A}, {$MACRO: A} и {$MACRO:"A"}. Такое поведение отлично от ключей элементов данных, где «key[a]», «key[a]» и «key["a"]» одинаковы семантически, но различны для критерия уникальности.

Секретные пользовательские макросы

Обзор

Zabbix предоставляет два варианта защиты конфиденциальной информации в значениях пользовательских макросов:

- Скрытый текст
- Секрет хранилища

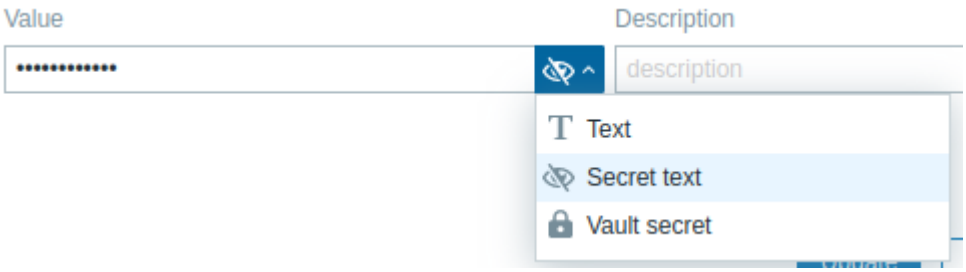
Note:
Хотя значение скрытого макроса и спрятано, его можно раскрыть посредством использования в элементах данных. Например, во внешнем скрипте оператор echo, ссылающийся на скрытый макрос, может использоваться для раскрытия значения макроса в веб-интерфейсе, поскольку сервер Zabbix имеет доступ к реальному значению макроса. Смотрите местоположения, где значения скрытых макросов не замаскированы.

Скрытые макросы нельзя использовать в выражениях триггеров.

Секретный текст

При использовании макросов Secret text значение макроса маскируется звездочками.

Чтобы сделать значение макроса секретным, нажмите кнопку в конце поля *Value* и выберите параметр *Secret text*:



После сохранения конфигурации просмотреть значение больше не будет возможно.

Чтобы изменить значение макроса, наведите курсор на поле *Value* и нажмите кнопку *Set new value* (появляется при наведении):



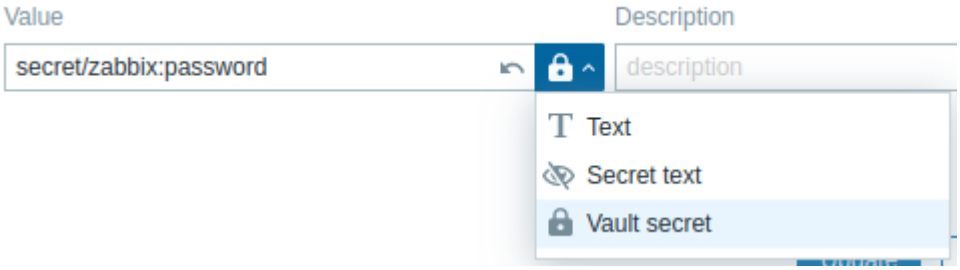
При нажатии кнопки *Set new value* (или при изменении типа значения макроса) текущее значение будет удалено. Вы можете восстановить исходное значение, нажав стрелку в конце поля *Value* (доступно только до сохранения новой конфигурации). Обратите внимание, что восстановление исходного значения не приведет к его раскрытию.

Note:
URL-адреса, содержащие секретный макрос, не будут работать, так как макрос в них будет преобразован в "******".

Секрет Vault

С помощью макросов секретов Vault значение макроса хранится во внешнем программном обеспечении для управления секретами (vault).

Чтобы настроить макрос секрета Vault, нажмите кнопку в конце поля *Value* и выберите опцию *Vault secret*:



Значение макроса должно указывать на секрет vault. Формат ввода зависит от поставщика vault. Примеры настройки для конкретных поставщиков см.:

- HashiCorp
- CyberArk

Значения макросов секретов Vault извлекаются из хранилища Zabbix server (и Zabbix proxy, если для параметра *Resolve secret vault macros by* задано значение *Zabbix server and proxy*) при каждом обновлении конфигурационных данных и затем сохраняются в кэше конфигурации.
Zabbix server и Zabbix proxy могут использовать разные хранилища.

Если для параметра *Resolve secret vault macros by* задано значение *Zabbix server*, то секреты Vault извлекаются только сервером, а Zabbix proxy получает значения макросов секретов Vault от Zabbix server при каждой синхронизации конфигурации и сохраняет их в собственном кэше конфигурации.
Это означает, что Zabbix proxy не может начать сбор данных после перезапуска, пока не получит обновление конфигурации от Zabbix server.

Чтобы вручную обновить значения секретов из хранилища, используйте параметр `secrets_reload` управления во время выполнения (только сервер).

Между Zabbix server и proxy должно быть включено шифрование; в противном случае в журнале будет записано предупреждающее сообщение сервера.

Warning:
Если значение макроса не удается успешно получить, соответствующий элемент данных, использующий это значение, перейдет в состояние unsupported.

Немаскированные расположения

Этот список содержит расположения параметров, в которых значения секретных макросов не маскируются.

Note:
Значения секретных макросов останутся замаскированными в указанных ниже расположениях, если на них ссылаются косвенно. Например, встроенные макросы {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY}, используемые в типах элементов данных (параметры Script или Webhook), будут разрешаться в ключи элементов данных, содержащие замаскированные секретные макросы, например `net.tcp.port[******,******]` вместо `net.tcp.port[192.0.2.1,80]`.

Контекст		Параметр
Элементы данных, прототипы элементов данных, правила LLD		
	Элемент данных	*Параметры кл*
	Прототип элемента данных	*Параметры кл*
	Правило низкоуровневого обнаружения	*Параметры кл*
	SNMP agent	*SNMP commun*
		Context name
		Security name
		Authentication
		Privacy passph
	HTTP agent	*URL*

Контекст		Параметр
		Query fields
		Request body
		Headers
		User name
		Password
		SSL key passwo
	Script	*Parameters*
		Script
	Browser	*Parameters*
		Script
	Database monitor	*SQL query*
	TELNET agent	*Script*
		User name
		Password
	SSH agent	*Script*
		User name
		Password
	Simple check	*User name*
		Password
	JMX agent	*User name*
		Password
Предварительная обработка значения элемента данных		
	Шаг предварительной обработки JavaScript	*Script*
Веб-сценарии		
	Веб-сценарий	*Variable value*
		Header value
		URL
		Query field valu
		Post field value
		Raw post
	Аутентификация веб-сценария	*User*
		Password
		SSL key passwo
Коннекторы		
	Коннектор	*URL*
		Username
		Password
		Token
		HTTP proxy
		SSL certificate
		SSL key file
		SSL key passwo
Сетевое обнаружение		
	SNMP	*SNMP commun*
		Context name
		Security name
		Authentication
		Privacy passph
Глобальные скрипты		
	Вебхук	*JavaScript scrip*
		JavaScript scrip
	Telnet	*Username*
		Password
	SSH	*Username*
		Password
	Script	*Script*
Типы медиа		
	Script	*Script paramet*
	Вебхук	*Parameters*
Управление IPMI		
	Узел сети	*Username*

Контекст	Параметр
	Password

Макросы низкоуровневого обнаружения

Обзор

Существует тип макроса, используемый внутри функции low-level discovery (LLD):

```
{#MACRO}
```

Это макрос, который используется в правиле LLD и возвращает реальные значения имени файловой системы, сетевого интерфейса, SNMP OID и т. д.

Эти макросы можно использовать для создания прототипов сущностей (прототипов элемента данных, триггера, графика, правила LLD, узла сети и группы узлов сети). Во время обнаружения реальных файловых систем, сетевых интерфейсов, виртуальных машин и т. д. эти макросы подставляются реальными значениями и служат основой для создания реальных сущностей.

Некоторые макросы low-level discovery поставляются "из коробки" вместе с функцией LLD в Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Однако при создании пользовательского правила low-level discovery следовать этим именам не обязательно. В этом случае можно использовать любое другое имя макроса LLD и ссылаться на него.

Поддерживаемые типы данных

При определении пользовательских правил обнаружения значения свойств, возвращаемые в объектах JSON для макросов LLD, должны быть одним из следующих примитивных типов:

- string
- number
- boolean

Массивы, объекты и нулевые значения не поддерживаются. Любой макрос LLD, ссылающийся на такое значение, во время предварительной обработки и создания элемента данных останется нераскрытым и будет отображаться буквально (например, `'{#MY_MACRO}'`).

Поддерживаемые места

LLD-макросы можно использовать:

- в фильтре правила низкоуровневого обнаружения
- во вложенных правилах низкоуровневого обнаружения, в
  - параметрах предварительной обработки JSONPath
  - поле JSONPath для пользовательских LLD-макросов
- для прототипов элементов данных и прототипов обнаружения в
  - имени
  - параметрах ключа
  - единице измерения
  - интервале обновления[1]
  - тайм-ауте[1]
  - периоде хранения истории[1]
  - периоде хранения трендов[1]
  - шагах предварительной обработки значения элемента данных
  - SNMP OID
  - поле датчика IPMI
  - выражении вычисляемого/агрегированного элемента данных, в:
    - константах выражения и параметрах функций
    - параметрах ключа элемента данных
  - условиях фильтра агрегированного элемента данных (имя группы узлов сети и имя тега)
  - скрипте SSH и скрипте Telnet
  - SQL-запросе мониторинга базы данных
  - поле endpoint элемента JMX
  - описании
  - поле URL агента HTTP
  - поле HTTP-запроса агента HTTP
  - поле тела запроса агента HTTP

    - поле требуемых кодов состояния агента HTTP
    - ключе и значении полей заголовков агента HTTP
    - поле имени пользователя для HTTP-аутентификации агента HTTP
    - поле пароля для HTTP-аутентификации агента HTTP
    - поле HTTP-прокси агента HTTP
    - поле файла HTTP SSL-сертификата агента HTTP
    - поле файла HTTP SSL-ключа агента HTTP
    - поле пароля HTTP SSL-ключа агента HTTP
    - тегах
- для прототипов триггеров в
    - имени
    - оперативных данных
    - выражении (только в константах и параметрах функций)
    - URL
    - описании
    - тегах
- для прототипов графиков в
    - имени
- для прототипов узлов сети в
    - имени
    - видимом имени
    - пользовательских полях интерфейса: IP, DNS, порт, SNMP v1/v2 community, имя контекста SNMP v3, имя безопасности SNMP v3, пароль аутентификации SNMP v3, пароль конфиденциальности SNMP v3
    - имени прототипа группы узлов сети
    - значении тега узла сети
    - значении макроса узла сети
    - (см. полный список)

Во всех этих местах, кроме фильтра правила низкоуровневого обнаружения, LLD-макросы можно использовать внутри статического контекста макроса.

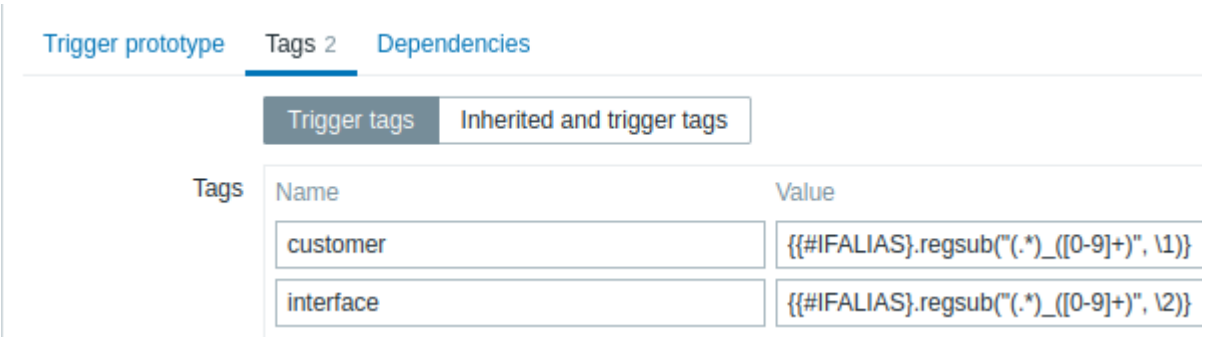
Использование функций макросов

Функции макросов поддерживаются макросами низкоуровневого обнаружения (за исключением фильтра правила низкоуровневого обнаружения), что позволяет извлекать определенную часть значения макроса с помощью регулярного выражения.

Например, для добавления тегов к событиям может потребоваться извлечь имя клиента и номер интерфейса из следующего макроса LLD:

```
{#IFALIAS}=customername_1
```

Для этого в поле значения тега события прототипа триггера можно использовать функцию макроса `regsub` вместе с макросом:



Обратите внимание, что запятые не допускаются в параметрах ключа элемента данных без кавычек, поэтому параметр, содержащий функцию макроса, необходимо заключить в кавычки. Для экранирования двойных кавычек внутри параметра следует использовать символ обратной косой черты (\). Пример:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Дополнительную информацию о синтаксисе функций макросов см. в разделе Функции макросов

Функции макросов поддерживаются в макросах низкоуровневого обнаружения начиная с Zabbix 4.0.

Прототипы элементов данных без параметров ключа

Когда параметры ключа элемента данных не используются, поместите макрос LLD внутри скобок параметра ключа элемента данных `[...]`, чтобы он обрабатывался как параметр и подставлялся во время обнаружения:

```
v_[{#MACRO}]
```

Примечания

1 В полях, отмеченных [1], один макрос должен заполнять целое поле. Несколько макросов в одном поле или макросы вперемешку с текстом не поддерживаются.

Макросы выражений

Обзор

Макросы выражений позволяют выполнять вычисления в полях.

Их значение вычисляется путем сначала разрешения всех вложенных макросов, а затем оценки полученного выражения.

Синтаксис:

```
{?EXPRESSION}
```

EXPRESSION использует тот же синтаксис и поддерживает те же функции, что и выражения триггеров.

Пример:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Для мест, где поддерживаются макросы выражений, найдите "{?EXPRESSION}" в таблице поддерживаемых макросов.

Примечания по использованию:

- Используйте макросы {FUNCTION.*}, чтобы ссылаться на значения функций выражений триггеров/выражений восстановления.
- Используйте макросы {HOST.HOST<1-9>} и {ITEM.KEY<1-9>}, чтобы ссылаться на узлы сети и элементы данных.
- В шаблонах используйте макросы {HOST.HOST<1-9>} или вообще не указывайте узел сети для первого узла — например, {?avg(//item1,1h)} — вместо имен шаблонов, так как имена шаблонов не заменяются именами узлов сети во время связывания шаблонов.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

См. также примеры выражений триггеров для примера использования макроса выражения в имени события.

Пользователи и группы пользователей

Обзор

Все пользователи в Zabbix получают доступ к приложению Zabbix через веб-интерфейс. Каждому пользователю назначаются уникальное имя для входа и пароль.

Все пароли пользователей шифруются и хранятся в базе данных Zabbix. Пользователи не могут использовать свой идентификатор пользователя и пароль для прямого входа на UNIX-сервер, если только они не были также соответствующим образом настроены в UNIX. Связь между веб-сервером и браузером пользователя может быть защищена с помощью SSL.

С помощью гибкой схемы прав пользователей вы можете ограничивать и разграничивать права на:

- Доступ к административным функциям веб-интерфейса Zabbix.
- Выполнение определенных действий в веб-интерфейсе.
- Доступ к контролируемым узлам сети в группах узлов сети.
- Использование определенных методов API.

Настройка пользователя

Обзор

В исходной установке Zabbix предусмотрены два пользователя:

- *Admin* — суперпользователь Zabbix с полным набором разрешений.
- *guest* — специальный пользователь Zabbix.

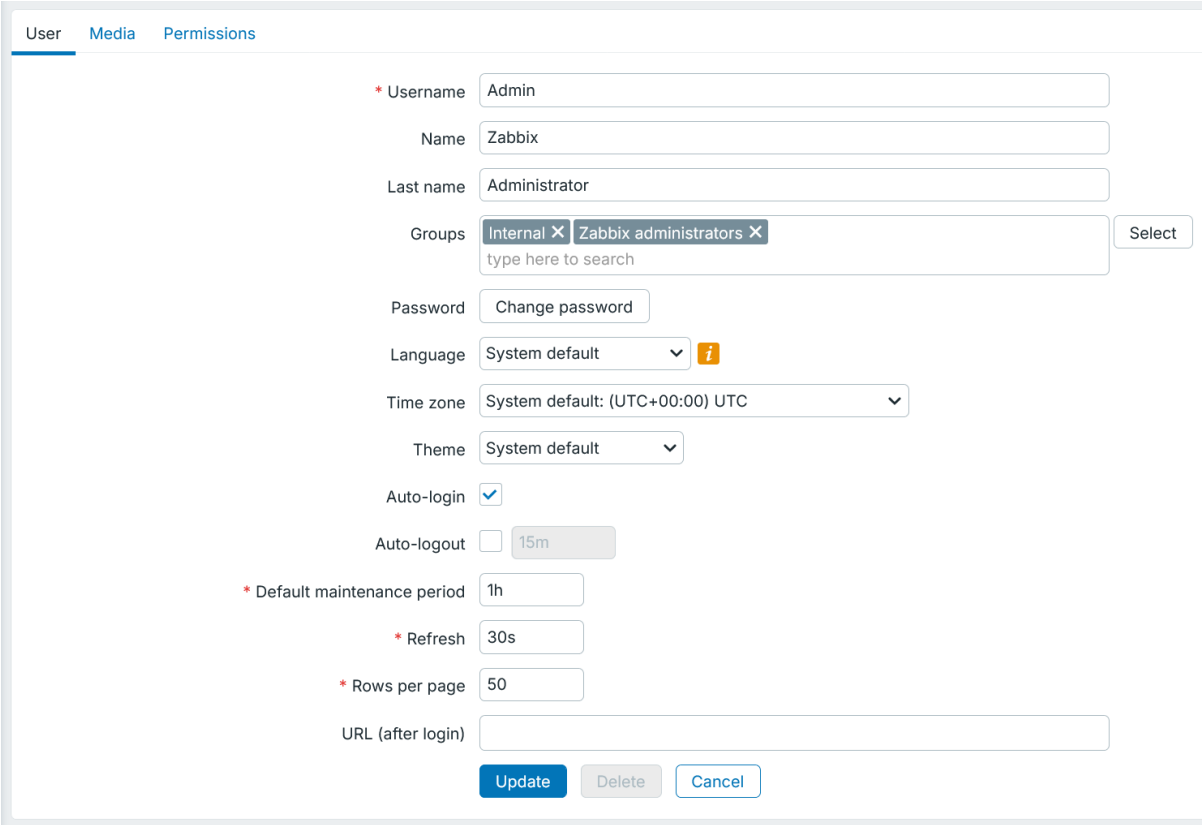
По умолчанию пользователь *guest* отключен.
Чтобы включить пользователя, удалите предопределенную группу *Disabled* из групп пользователей.
Вы можете войти в систему с этим пользователем и получить доступ к страницам мониторинга в Zabbix.
Обратите внимание, что по умолчанию у пользователя *guest* нет разрешений на объекты Zabbix.

Чтобы настроить пользователя:

1. Перейдите в раздел *Пользователи* > *Пользователи*.
2. Нажмите *Создать пользователя* (или выберите имя пользователя, чтобы изменить существующего пользователя).
3. Измените атрибуты пользователя в форме.

Общие атрибуты

Вкладка *Пользователь* содержит общие атрибуты пользователя:



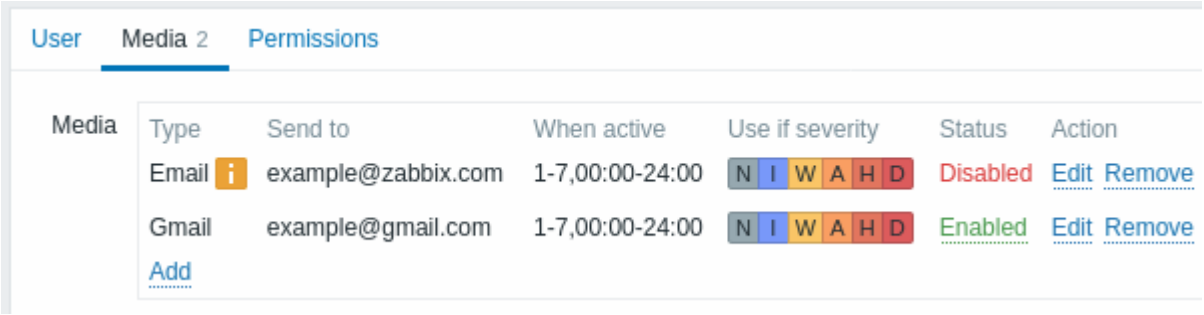
Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя пользователя	Уникальное имя пользователя, используемое в качестве имени для входа.
Имя	Имя пользователя (необязательно). Если поле заполнено, имя отображается в информации о подтверждении и в информации о получателе уведомления.
Фамилия	Фамилия пользователя (необязательно). Если поле заполнено, фамилия отображается в информации о подтверждении и в информации о получателе уведомления.
Группы	Выберите группы пользователей, в которые входит пользователь. Это поле поддерживает автодополнение: начните вводить имя группы пользователей, и в раскрывающемся списке появятся соответствующие группы. Прокрутите список вниз, чтобы выбрать группу. Также можно нажать *Выбрать*, чтобы добавить группы. Нажмите «x», чтобы удалить выбранную группу. Членство в группах пользователей определяет, к каким группам узлов сети и узлам сети пользователь будет иметь доступ.

Параметр	Описание
Пароль	Два поля для ввода пароля пользователя или кнопка *Изменить пароль*, если пользователь уже существует. При нажатии кнопки *Изменить пароль* открываются два поля для ввода нового пароля. Для пользователя с ролью *Суперадминистратор*, изменяющего собственный пароль, при нажатии кнопки *Изменить пароль* открывается дополнительное поле для ввода текущего (старого) пароля. После успешного изменения пароля пользователь, для которого был изменен пароль, будет отключен от всех активных сеансов. Обратите внимание, что пароль можно изменить только для пользователей, использующих внутреннюю аутентификацию Zabbix.
Язык	Язык веб-интерфейса Zabbix. Для работы переводов требуется расширение php gettext.
Часовой пояс	Выберите часовой пояс, чтобы переопределить глобальный часовой пояс на уровне пользователя, или выберите **Системный по умолчанию**, чтобы использовать глобальные настройки часового пояса.
Тема	Определяет внешний вид веб-интерфейса: **Системная по умолчанию** - использовать системные настройки по умолчанию **Синяя** - стандартная синяя тема **Синяя (классическая)** - стандартная синяя тема с использованием устаревших шрифтов **Темная** - альтернативная темная тема **Темно-синяя** - темно-синяя тема **Темная (классическая)** - альтернативная темная тема с использованием устаревших шрифтов **Светлая с высокой контрастностью** - светлая тема с высокой контрастностью **Темная с высокой контрастностью** - темная тема с высокой контрастностью
Автоматический вход	Установите этот флажок, чтобы Zabbix запоминал пользователя и автоматически выполнял вход в течение 30 дней. При входе с параметром *Запомнить на 30 дней*: - Автоматический выход сбрасывается (сеанс сохраняется в течение 30 дней). - Автоматический вход включается для повторной аутентификации без дополнительных действий. При входе без параметра *Запомнить на 30 дней*: - Автоматический вход включается для повторной аутентификации без дополнительных действий. - Автоматический выход по-прежнему регулируется стандартным значением тайм-аута. Для этого используются файлы cookie браузера.
Автоматический выход	Если этот флажок установлен, пользователь будет автоматически отключен по истечении заданного количества секунд (минимум 90 секунд, максимум 1 день). Обратите внимание, что этот параметр переопределяется, если включен параметр *Запомнить на 30 дней*, поскольку сеанс продлевается на весь этот период. Поддерживаются суффиксы времени, например: 90s, 5m, 2h, 1d. Обратите внимание, что этот параметр не будет работать: * Если включен глобальный параметр конфигурации «Показывать предупреждение, если сервер Zabbix не работает», а веб-интерфейс Zabbix остается открытым. * Когда страницы меню Мониторинг выполняют обновление информации в фоновом режиме. * Если вход выполнен с установленным параметром *Запомнить меня на 30 дней*.
Период обслуживания по умолчанию	Задайте период обслуживания по умолчанию для внепланового обслуживания.
Обновление	Задайте частоту обновления, используемую для графиков, данных в виде обычного текста и т. д. Можно установить значение 0, чтобы отключить обновление. Поддерживаются суффиксы времени, например: 90s, 5m, 1h.
Строк на странице	Можно определить количество строк, отображаемых на странице в списках.
URL (после входа)	Можно настроить перенаправление пользователя Zabbix на определенный URL после успешного входа, например на страницу *Проблемы*.

Пользовательские способы оповещения

Вкладка *Способы оповещения* содержит список всех способов оповещения, определенных для пользователя. Способы оповещения используются для отправки уведомлений.



Нажмите *Добавить*, чтобы назначить способ оповещения пользователю.

Если тип способа оповещения отключен:

- после его названия отображается желтый информационный значок;
- в столбце «Статус» отображается *Отключен*.

Note:
Разрешения пользователя на изменение сведений о собственных способах оповещения можно предоставлять или отзывать на основе его роли пользователя (см. параметр *Создание и редактирование собственных способов оповещения*). Разрешения суперпользователя на изменение сведений о способах оповещения других пользователей также можно предоставлять или отзывать на основе его роли пользователя (см. параметр *Создание и редактирование способов оповещения пользователей*).

Подробнее о настройке пользовательских способов оповещения см. в разделе Типы способов оповещения.

Права доступа

Вкладка *Права доступа* содержит информацию о следующих элементах:

- Роль пользователя (обязательна для любого нового пользователя), которую может изменить только пользователь типа *Суперадминистратор*.

Attention:
Пользователей нельзя создавать без роли пользователя (за исключением использования User API Zabbix). Ранее созданных пользователей, у которых нет роли, можно редактировать без назначения им роли. Однако после назначения роли ее можно только изменить, но не удалить.

 Обратите внимание, что пользователи без роли могут войти в Zabbix только с использованием аутентификации LDAP или SAML, при условии, что их данные LDAP/SAML соответствуют сопоставлениям групп пользователей, настроенным в Zabbix.

- Тип пользователя (*Пользователь*, *Администратор*, *Суперадминистратор*), определяемый в настройках роли пользователя.
- Группы узлов сети и шаблонов, к которым пользователь имеет доступ.
  - Пользователи типов *Пользователь* и *Администратор* по умолчанию не имеют доступа ни к каким группам, шаблонам и узлам сети. Чтобы предоставить такой доступ, пользователей необходимо включить в группы пользователей, для которых настроены права доступа к соответствующим объектам.
- Права доступа к разделам и элементам веб-интерфейса Zabbix, модулям и методам API.
  - Элементы, доступ к которым разрешен, отображаются зеленым цветом, а элементы с запрещенным доступом - светло-серым.
  - Если доступ к API включен, методы API роли пользователя перечисляются в разделе *Разрешенные методы* или *Запрещенные методы* в зависимости от того, что выбрано в настройках роли пользователя: *Список разрешений* или *Список запретов*. Если методы API не указаны, отображается *Нет*. Значение *Нет* в разделе *Разрешенные методы* означает, что ни один метод API не разрешен, а значение *Нет* в разделе *Запрещенные методы* означает, что разрешены все методы API, доступные для данного типа пользователя.
- Права на выполнение определенных действий.
  - Действия, которые пользователю разрешено выполнять, отображаются зеленым цветом, а запрещенные действия - светло-серым.

Подробнее см. на странице Права доступа.

Разрешения

Обзор

Права доступа пользователей в Zabbix зависят от типа пользователя, настраиваемых ролей пользователя и доступа к узлам сети, который задается для группы пользователей.

Типы пользователей

Права доступа в Zabbix в первую очередь зависят от типа пользователя:

- *Пользователь* — имеет ограниченные права доступа к разделам меню (см. ниже) и по умолчанию не имеет доступа ни к каким ресурсам. Любые права доступа к группам узлов сети или шаблонов необходимо назначать явно.
- *Администратор* — имеет неполные права доступа к разделам меню (см. ниже). По умолчанию пользователь не имеет доступа ни к одной группе узлов сети. Любые права доступа к группам узлов сети или шаблонов необходимо назначать явно.
- *Суперадминистратор* — имеет доступ ко всем разделам меню. Пользователь имеет доступ на чтение и запись ко всем группам узлов сети и шаблонов. Права доступа нельзя отозвать путем запрета доступа к определенным группам.

Доступ к меню

> **Note:**
> Ограничение доступа только к некоторым элементам веб-интерфейса предотвращает открытие этой страницы, но не лишает возможности получить доступ к базовым данным в других разделах интерфейса.

В следующей таблице показан доступ к разделам меню Zabbix для каждого типа пользователя:

Раздел меню		Пользователь	Администратор	Суперадминистратор
Панели		+	+	+
Мониторинг		+	+	+
	Проблемы	+	+	+
	Узлы сети	+	+	+
	Последние данные	+	+	+
	Карты	+	+	+
	Обнаружение		+	+
APM		+	+	+
	Трассировки	+	+	+
	Метрики	+	+	+
	Журналы	+	+	+
Сервисы		+	+	+
	Сервисы	+	+	+
	SLA		+	+
	Отчет SLA	+	+	+
Инвентаризация		+	+	+
	Обзор	+	+	+
	Узлы сети	+	+	+
Отчеты		+	+	+
	Сведения о системе		+	+
	Запланированные отчеты		+	+
	Отчет о доступности	+	+	+
	100 лучших триггеров	+	+	+
	Журнал аудита			+
	Журнал действий			+
	Уведомления		+	+
Сбор данных			+	+
	Группы шаблонов		+	+
	Группы узлов сети		+	+
	Шаблоны		+	+
	Узлы сети		+	+
	Обслуживание		+	+
	Корреляция событий			+
	Обнаружение		+	+
Оповещения			+	+
	Действия с триггерами		+	+
	Действия с сервисами		+	+
	Действия обнаружения		+	+
	Действия авторегистрации		+	+

Раздел меню		Пользователь	Администратор	Суперадминистратор
	Внутренние действия		+	+
	Типы носителей			+
	Скрипты			+
Пользователи				+
	Группы пользователей			+
	Роли пользователей			+
	Пользователи			+
	Токены API			+
	Аутентификация			+
	Устройства			+
Администрирование				+
	Источник данных			+
	Общие параметры			+
	Журнал аудита			+
	Очистка данных			+
	Группы прокси			+
	Прокси			+
	Макросы			+
	Очередь			+

Роли пользователей

Роли пользователей позволяют настраивать разрешения, определенные типом пользователя. Хотя добавлять разрешения нельзя (это превысило бы разрешения типа пользователя), некоторые разрешения можно отозвать.

Кроме того, роль пользователя определяет доступ не только к разделам меню, но и к сервисам, модулям, методам API и различным действиям в веб-интерфейсе.

Роли пользователей настраиваются суперпользователями в разделе *Пользователи* > *Роли пользователей*.

Роли пользователей назначаются пользователям суперпользователями в форме настройки пользователя, на вкладке *Разрешения*.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

Доступ к узлам сети

Доступ к любым данным узлов сети и шаблонов в Zabbix назначается группам пользователей только на уровне групп узлов сети/шаблонов.

Это означает, что отдельному пользователю нельзя напрямую предоставить разрешение на доступ к какому-либо узлу сети (или к группе узлов сети). Пользователь может получить доступ к узлу сети, только будучи частью группы пользователей, которая имеет доступ к группе узлов сети, содержащей необходимый узел сети.

Таким же образом пользователю можно предоставить доступ к шаблону, только включив его в группу пользователей, имеющую доступ к группе шаблонов, в которую входит соответствующий шаблон.

Группы пользователей

Обзор

Группы пользователей позволяют объединять пользователей как в организационных целях, так и для назначения разрешений на доступ к данным. Разрешения на просмотр и настройку данных групп узлов сети и групп шаблонов назначаются группам пользователей, а не отдельным пользователям.

Часто имеет смысл разделить информацию, доступную одной группе пользователей, и информацию, доступную другой группе. Этого можно добиться, объединив пользователей в группы, а затем назначив различные разрешения на группы узлов сети и группы шаблонов.

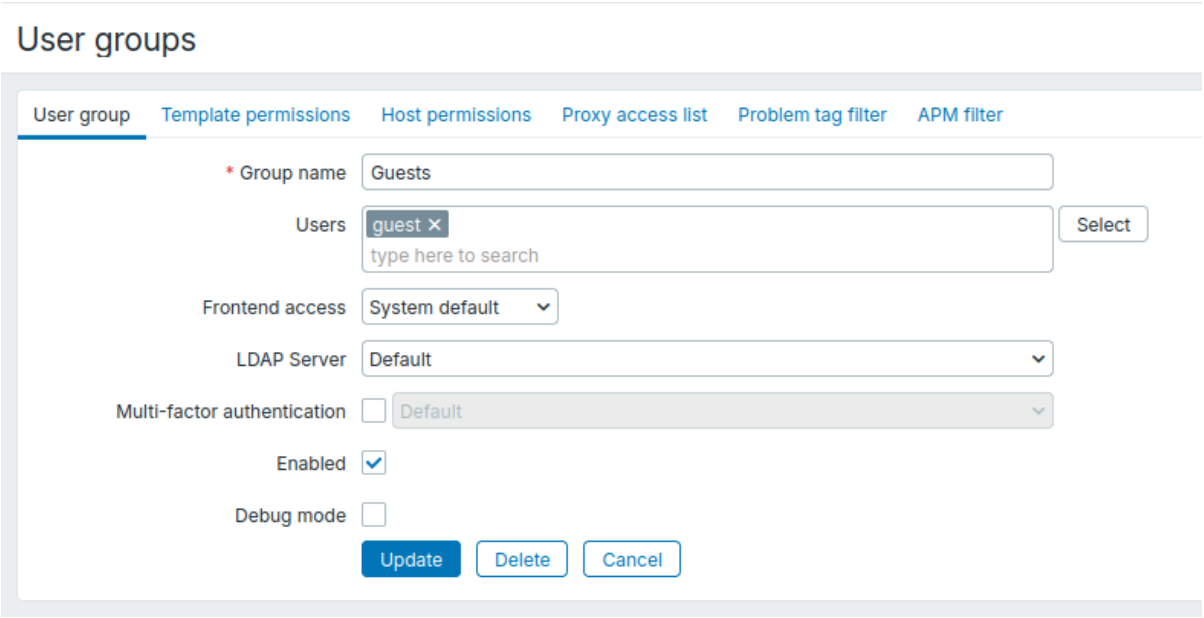
Пользователь может входить в любое количество групп.

Настройка

Чтобы настроить группу пользователей:

1. Перейдите в *Пользователи* > *Группы пользователей*.
2. Нажмите *Создать группу пользователей* (или имя группы, чтобы изменить существующую группу).
3. Измените атрибуты группы в форме.

Вкладка **Группа пользователей** содержит общие атрибуты группы:

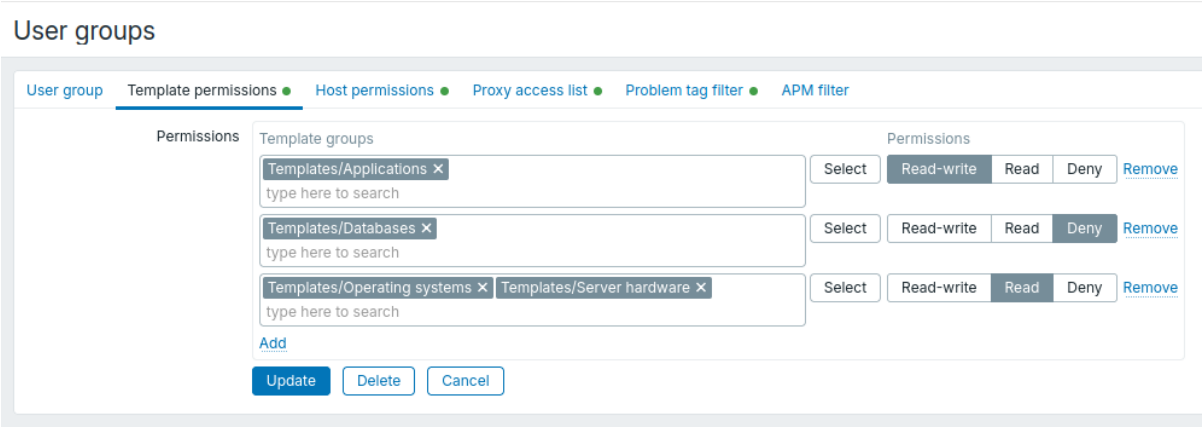


Все обязательные поля ввода отмечены красной звездочкой.

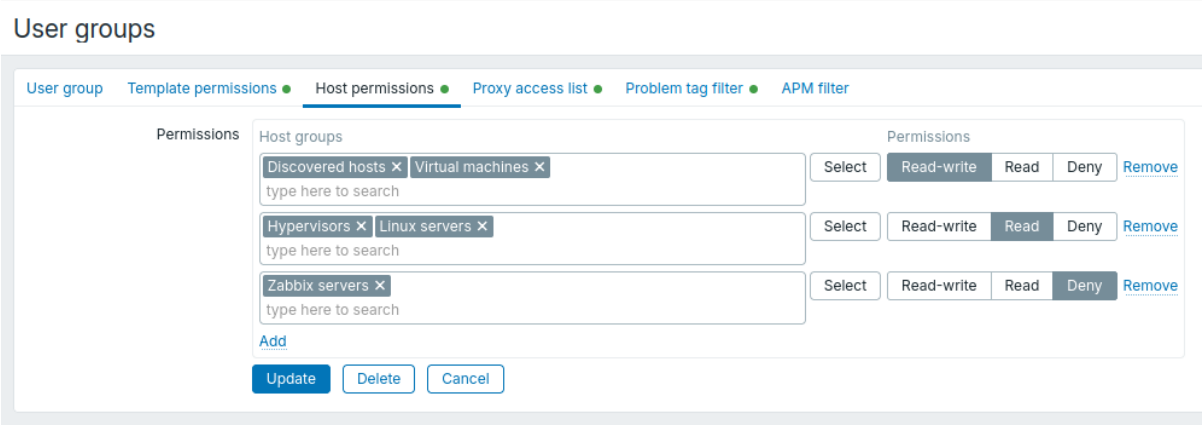
Параметр	Описание
Имя группы	Уникальное имя группы.
Пользователи	Чтобы добавить пользователей в группу, начните вводить имя существующего пользователя. Когда появится раскрывающийся список с подходящими именами пользователей, прокрутите его вниз и выберите нужного пользователя. Также можно нажать кнопку *Выбрать*, чтобы выбрать пользователей во всплывающем окне.

Параметр	Описание
Доступ к веб-интерфейсу	Способ аутентификации пользователей группы. **По умолчанию** - использовать метод аутентификации по умолчанию (заданный глобально) **Внутренняя** - использовать внутреннюю аутентификацию Zabbix (даже если глобально используется аутентификация LDAP). Игнорируется, если глобальным методом по умолчанию является аутентификация HTTP. **LDAP** - использовать аутентификацию LDAP (даже если глобально используется внутренняя аутентификация). Игнорируется, если глобальным методом по умолчанию является аутентификация HTTP. **Отключен** - доступ к веб-интерфейсу Zabbix для этой группы запрещен
Сервер LDAP	Выберите, какой сервер LDAP использовать для аутентификации пользователя. Это поле доступно только если для параметра *Доступ к веб-интерфейсу* выбрано значение LDAP или По умолчанию.
Многофакторная аутентификация	Выберите, какой метод многофакторной аутентификации использовать для аутентификации пользователя: **По умолчанию** - использовать метод, заданный по умолчанию в настройках MFA; этот вариант выбран по умолчанию для новых групп пользователей, если MFA включена; **<Имя метода>** - использовать выбранный метод (например, «Zabbix TOTP»); **Отключена** - MFA отключена для этой группы; этот вариант выбран по умолчанию для новых групп пользователей, если MFA отключена. Обратите внимание: если пользователь входит в несколько групп пользователей с включенной MFA (или хотя бы в одной группе MFA включена), применяются следующие правила аутентификации: если какая-либо группа использует метод MFA «По умолчанию», пользователь будет аутентифицирован с его помощью; в противном случае для аутентификации будет использован первый метод в алфавитном порядке.
Включена	Состояние группы пользователей и ее участников. *Установлен* - группа пользователей и пользователи включены *Снят* - группа пользователей и пользователи отключены
Режим отладки	Установите этот флажок, чтобы активировать режим отладки для пользователей.

Вкладка **Разрешения шаблонов** позволяет указать доступ группы пользователей к группам шаблонов и через них - к данным шаблонов:



Вкладка **Разрешения узлов сети** позволяет указать доступ группы пользователей к группам узлов сети и через них - к данным узлов сети:



Нажмите Add, чтобы выбрать группы шаблонов/узлов сети (родительскую или вложенную группу) и назначить им разрешения. Начните вводить имя группы (появится раскрывающийся список подходящих групп) или нажмите *Выбрать*, чтобы открыть всплывающее окно со списком всех групп.

Затем с помощью кнопок выбора назначьте разрешения выбранным группам. Доступны следующие разрешения:

- **Чтение-запись** - доступ к группе с правами чтения и записи
- **Чтение** - доступ к группе только для чтения
- **Запретить** - доступ к группе запрещен

Если одна и та же группа шаблонов/узлов сети добавлена в несколько строк с разными разрешениями, будет применено наиболее строгое разрешение.

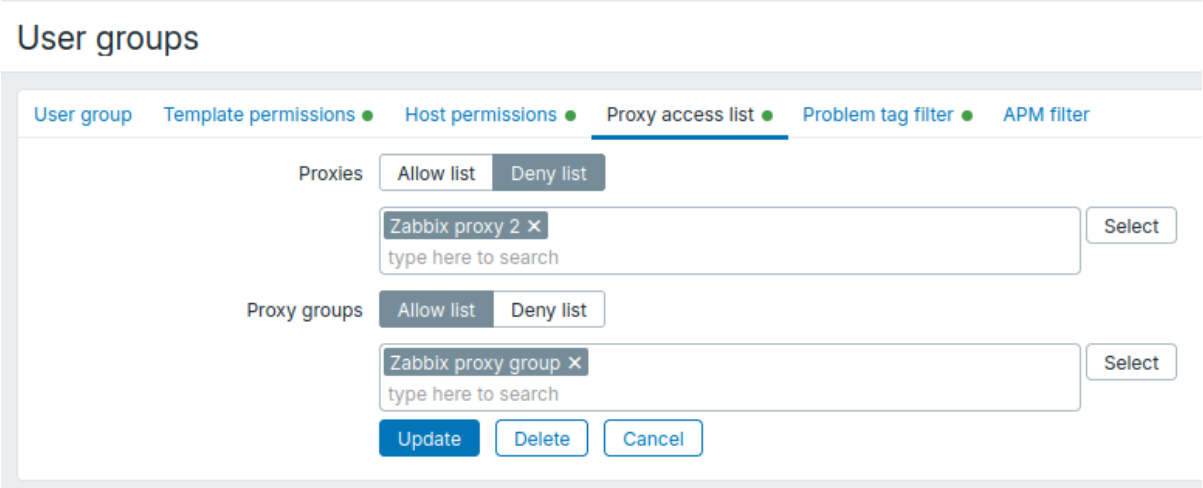
Обратите внимание, что пользователь с ролью *Суперадминистратор* может принудительно назначить вложенным группам тот же уровень разрешений, что и у родительской группы; это можно сделать в форме настройки группы узлов сети/шаблонов.

Вкладки **Разрешения шаблонов** и **Разрешения узлов сети** поддерживают одинаковый набор параметров.

Текущие разрешения для групп отображаются в блоке *Разрешения*, где их можно изменить или удалить.

Note:
Если группа пользователей имеет разрешения **Чтение-запись** для узла сети и разрешение **Запретить** или не имеет разрешений для шаблона, связанного с этим узлом сети, пользователи такой группы не смогут изменять элементы данных, созданные на основе шаблона, на узле сети, а имя шаблона будет отображаться как *Недоступный шаблон*.

Вкладка **Список доступа к прокси** позволяет указать доступ группы пользователей к прокси - отдельно или через группы прокси:



Используйте поле *Группы прокси*, чтобы применить разрешение к выбранным группам. Прокси, принадлежащий группе, можно разрешить или запретить только через эту группу. Используйте поле *Прокси*, чтобы добавить отдельные прокси, не принадлежащие ни одной группе прокси.

Поля *Прокси* и *Группы прокси* поддерживают одинаковый набор параметров.

Нажмите кнопку *Выбрать* рядом с полем *Прокси* или *Группы прокси*, чтобы выбрать прокси или группы прокси соответственно. Также можно начать вводить имя прокси или группы прокси в соответствующем поле. Появится раскрывающийся список подходящих прокси или групп прокси.

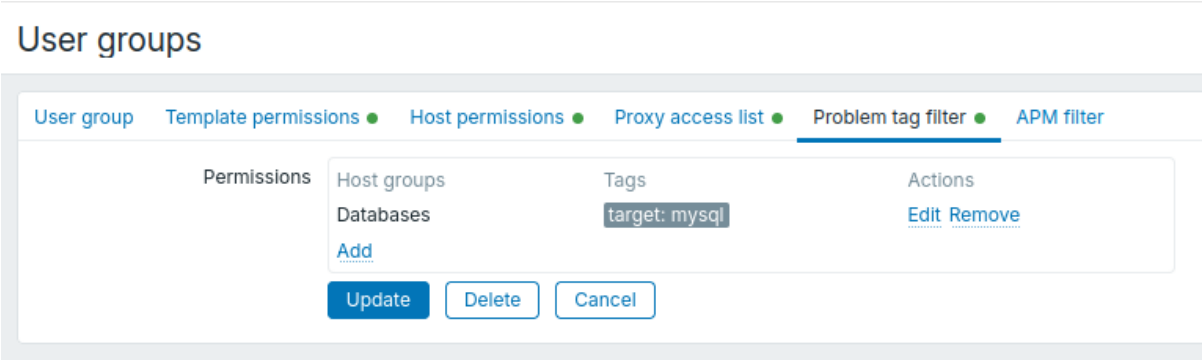
Выберите *Список разрешенных*, чтобы разрешить группе пользователей доступ только к выбранным прокси и группам прокси.
 Если в поле *Прокси* для списка разрешенных ничего не выбрано, группа пользователей не будет иметь доступа к прокси, не принадлежащим группе прокси. Если в поле *Группы прокси* для списка разрешенных ничего не выбрано, группа пользователей не будет иметь доступа ни к одной группе прокси.

Выберите *Список запрещенных*, чтобы запретить группе пользователей доступ к выбранным прокси и группам прокси. Доступ ко всем остальным прокси и группам прокси разрешен.
 Если в поле *Прокси* для списка запрещенных ничего не выбрано, группа пользователей будет иметь полный доступ ко всем прокси, не принадлежащим группе прокси. Если в поле *Группы прокси* для списка запрещенных ничего не выбрано, группа пользователей будет иметь полный доступ ко всем группам прокси.

Правила приоритета разрешений для пользователей, входящих в несколько групп пользователей:

- *Список запрещенных* имеет приоритет над *Списком разрешенных*. Если какая-либо группа запрещает доступ к прокси или группе прокси, пользователь не сможет получить к ним доступ.
- Разрешения *Списка разрешенных* суммируются. Если несколько групп разрешают доступ, пользователь сможет получить доступ ко всем прокси, разрешенным этими группами.

Вкладка **Фильтр тегов проблем** позволяет настроить разрешения на основе тегов, чтобы группы пользователей могли просматривать проблемы, отфильтрованные по имени и значению тега:

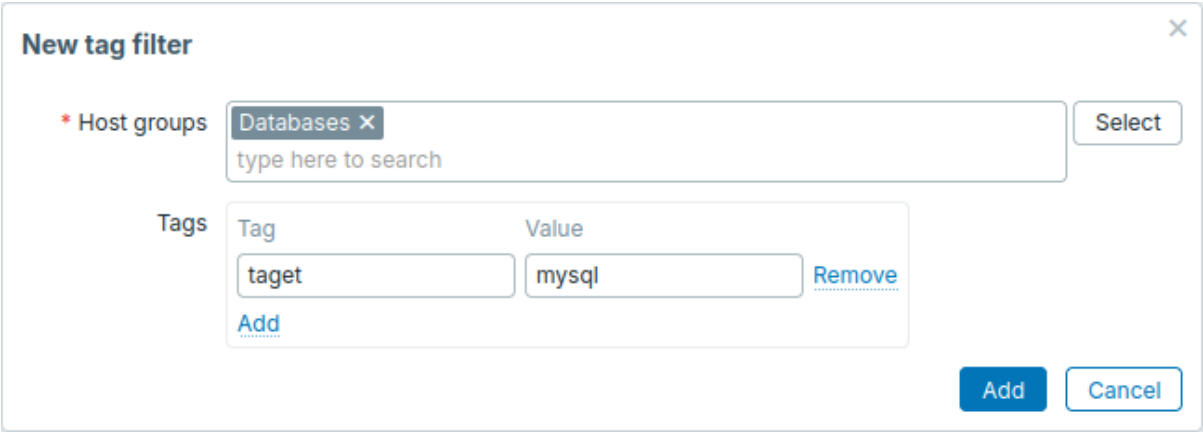


Нажмите Add, чтобы выбрать группы узлов сети. Чтобы выбрать группу узлов сети, для которой нужно применить фильтр тегов, нажмите *Выбрать*, чтобы получить полный список существующих групп узлов сети, или начните вводить имя группы узлов сети, чтобы открыть раскрывающийся список подходящих групп. Отображаться будут только группы узлов сети, поскольку фильтр тегов проблем нельзя применить к группам шаблонов.

Теги без значений можно добавлять, но значения без имен добавлять нельзя. В блоке *Разрешения* отображаются только первые три тега (при наличии значений); если тегов больше, их можно просмотреть, нажав на значок ••• или наведя на него указатель мыши.

Фильтр тегов позволяет отделить доступ к группе узлов сети от возможности просматривать проблемы.

Например, если администратору базы данных необходимо видеть только проблемы базы данных «MySQL», сначала нужно создать группу пользователей для администраторов баз данных, а затем указать имя тега «target» и значение «mysql».



Если указано имя тега «target», а поле значения оставлено пустым, группа пользователей будет видеть все проблемы с именем тега «target» для выбранной группы узлов сети.

Убедитесь, что имя и значение тега указаны правильно, иначе группа пользователей не будет видеть никаких проблем.

Рассмотрим пример, когда пользователь является участником нескольких выбранных групп пользователей. В этом случае для тегов будет использоваться условие ИЛИ.

Группа пользователей A			**Группа пользователей B**			**Видимый результат для пользователя (участника) обеих групп**
Фильтр тегов						
Группа узлов сети	*Имя тега*	*Значение тега*	*Группа узлов сети*	*Имя тега*	*Значение тега*	
Базы данных	target	mysql	Базы данных	target	oracle	Видны проблемы target:mysql или target:oracle
Не настроено в разделе **Фильтр тегов проблем**			Базы данных	target	oracle	Видны проблемы target:oracle

Доступ из нескольких групп пользователей

Пользователь может принадлежать к любому количеству групп пользователей. Эти группы могут иметь разные права доступа к узлам сети или шаблонам.

Поэтому важно понимать, к каким объектам сможет получить доступ непривилегированный пользователь в результате. В следующем примере рассмотрите, как доступ к узлу сети **X** (в Hostgroup 1) будет изменяться в различных ситуациях для пользователя, который состоит в группах пользователей A и B.

- Если у Group A есть только доступ *Read* к Hostgroup 1, а у Group B есть доступ *Read-write* к Hostgroup 1, пользователь получит доступ **Read-write** к 'X'.

Attention:
Права *Read-write* имеют приоритет над правами *Read*.

- В том же сценарии, что и выше, если 'X' одновременно также входит в Hostgroup 2, в котором доступ для Group A или B **запрещен**, доступ к 'X' будет **недоступен**, несмотря на доступ *Read-write* к Hostgroup 1.
- Если у Group A не определены права, а у Group B есть доступ *Read-write* к Hostgroup 1, пользователь получит доступ **Read-write** к 'X'.
- Если у Group A есть доступ *Deny* к Hostgroup 1, а у Group B есть доступ *Read-write* к Hostgroup 1, доступ пользователя к 'X' будет **запрещен**.

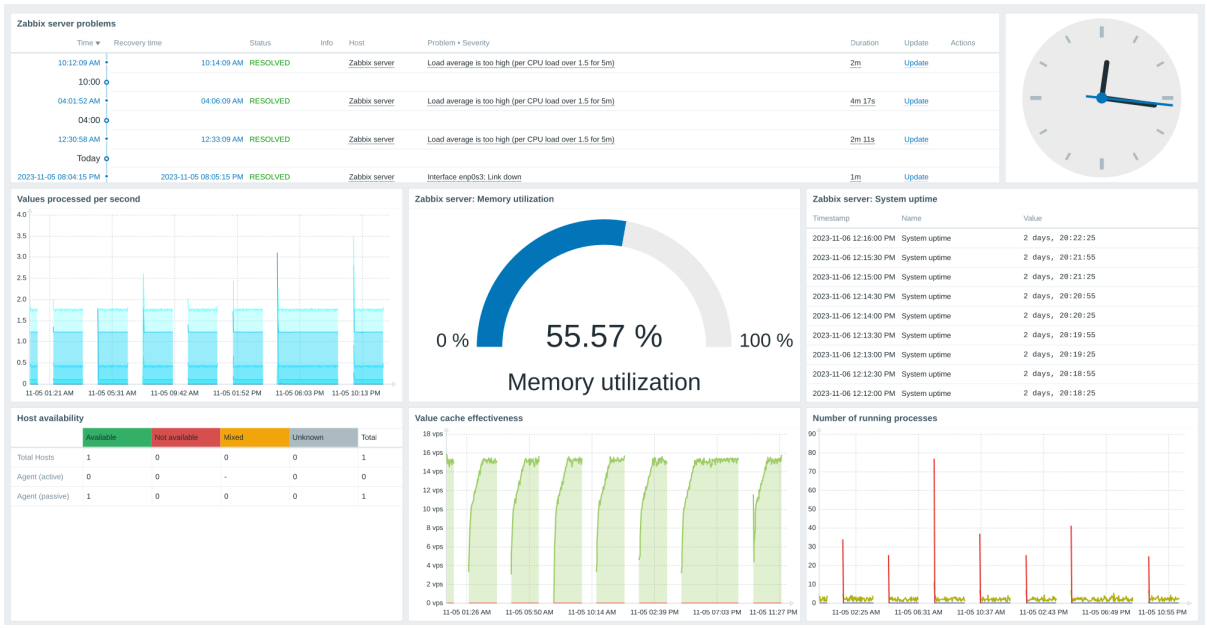
Другие сведения

- Пользователь уровня Admin с доступом *Read-write* к узлу сети не сможет связывать/отвязывать шаблоны, если у него нет доступа к группе шаблонов, к которой они относятся. При доступе *Read* к группе шаблонов он сможет связывать/отвязывать шаблоны с узлом сети, однако не будет видеть никаких шаблонов в списке шаблонов и не сможет выполнять операции с шаблонами в других местах.
- Пользователь уровня Admin с доступом *Read* к узлу сети не будет видеть узел сети в списке узлов сети в разделе конфигурации; однако триггеры узла сети будут доступны в конфигурации IT service.
- Любой пользователь, не являющийся Super Admin (включая 'guest'), может просматривать карты сети, если карта пуста или содержит только изображения. Когда на карту добавляются узлы сети, группы узлов сети или триггеры, учитываются права доступа.
- Сервер Zabbix не будет отправлять уведомления пользователям, указанным в качестве получателей операций действия, если доступ к соответствующему узлу сети явно "denied".

Запланированные отчеты

Обзор

С помощью функции *Запланированные отчёты* можно настроить отправку PDF-версии определённой панели мониторинга указанным получателям через регулярные интервалы.



Предварительные требования:

- Веб-сервис Zabbix должен быть установлен и настроен надлежащим образом для включения создания запланированных отчётов — инструкции см. в разделе Настройка запланированных отчётов.
- Пользователь должен иметь роль пользователя типа *Администратор* или *Суперадминистратор* со следующими разрешениями:
  - *Запланированные отчёты* в блоке *Доступ к элементам веб-интерфейса* (для просмотра настроек отчётов)
  - *Управление запланированными отчётами* в блоке *Доступ к действиям* (для создания и редактирования отчётов)

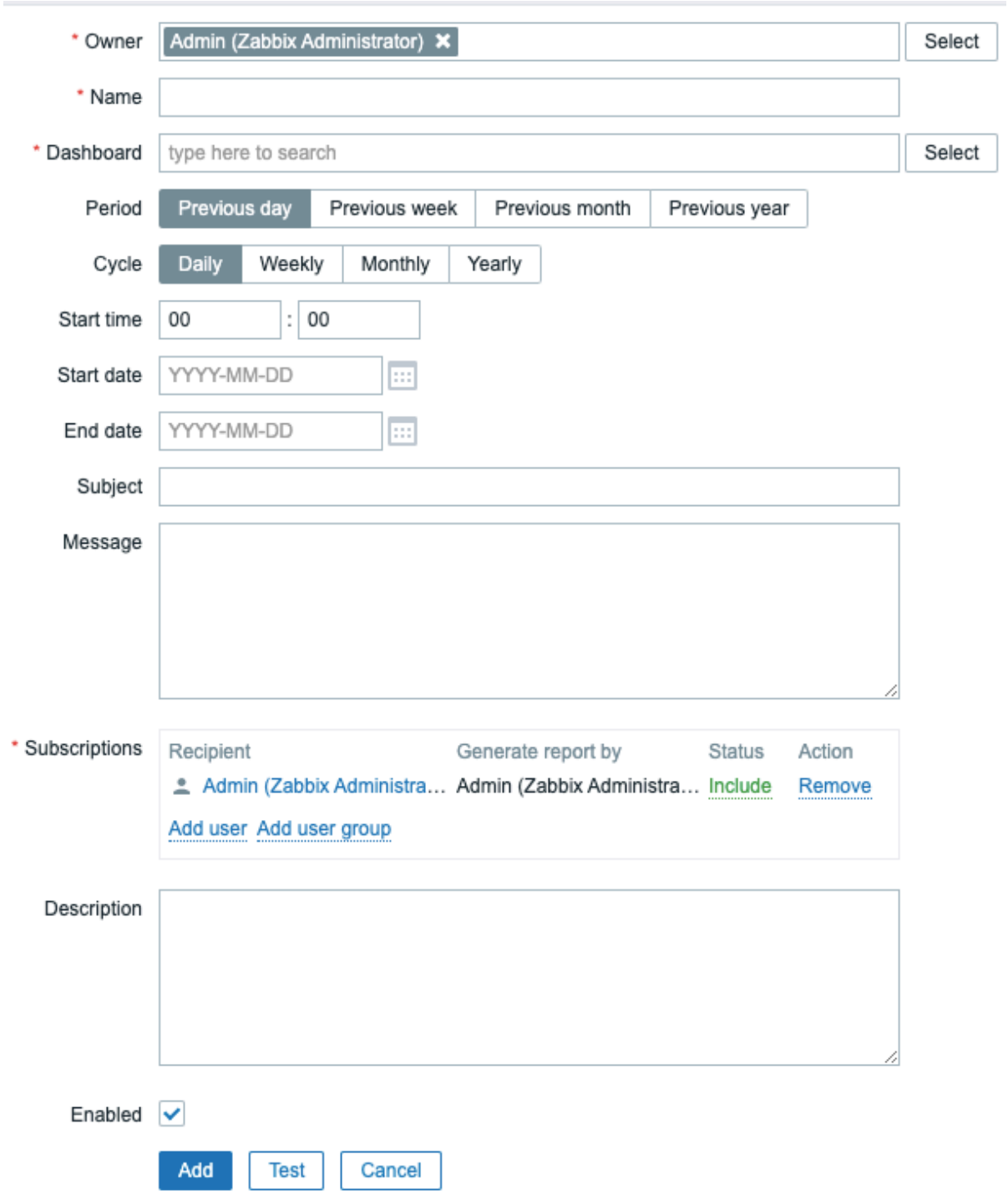
Чтобы создать запланированный отчёт в веб-интерфейсе Zabbix, выполните следующие действия:

1. Перейдите в раздел *Отчёты* > *Запланированные отчёты*.
2. Нажмите *Создать отчёт* в правом верхнем углу экрана.
3. Укажите параметры отчёта в форме.

Также можно создать отчёт, открыв существующий отчёт, нажав кнопку *Клонировать* и сохранив его под другим именем.

Настройка

Вкладка *Запланированные отчеты* содержит общие атрибуты отчета.



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Владелец	Пользователь, создающий отчет. Пользователи с уровнем *Суперадминистратор* могут изменять владельца. Для пользователей с уровнем *Администратор* это поле доступно только для чтения.
Имя	Имя отчета; должно быть уникальным.
Панель мониторинга	Панель мониторинга, на основе которой создается отчет; одновременно можно выбрать только одну панель мониторинга. Чтобы выбрать панель мониторинга, начните вводить ее имя — появится список соответствующих панелей мониторинга; прокрутите список вниз, чтобы выбрать нужную. Также можно нажать *Выбрать* рядом с полем и выбрать панель мониторинга из отображаемого списка.
Период	Период времени, за который будет подготовлен отчет. Выберите предыдущий день, неделю, месяц или год.
Цикл	Частота создания отчетов. Отчеты можно отправлять ежедневно, еженедельно, ежемесячно или ежегодно. В режиме «Еженедельно» можно выбрать дни недели, в которые будет отправляться отчет.

Параметр	Описание
Время начала	Время суток в формате чч:мм, когда будет подготовлен отчет. Обратите внимание, что будет использоваться часовой пояс сервера Zabbix.
Повторять в	Дни недели, в которые будет отправляться отчет. Это поле доступно только если для параметра *Цикл* установлено значение «Еженедельно».
Дата начала	Дата начала регулярного создания отчетов.
Дата окончания	Дата окончания регулярного создания отчетов.
Тема	Тема электронного письма с отчетом. Поддерживаемые макросы: {TIME}, {TIMESTAMP}.
Сообщение	Текст электронного письма с отчетом. Поддерживаемые макросы: {TIME}, {TIMESTAMP}.
Подписки	Список получателей отчета. По умолчанию включает только владельца отчета. Получателем отчета можно указать любого пользователя Zabbix с настроенным каналом электронной почты. Нажмите *Добавить пользователя* или *Добавить группу пользователей*, чтобы добавить получателей. Нажмите имя пользователя, чтобы изменить настройки: *Формировать отчет от имени* — следует ли формировать данные отчета на основе разрешений текущего пользователя или получателя на панели мониторинга. *Статус* — выберите «Включить», чтобы отправлять отчет пользователю, или «Исключить», чтобы не отправлять отчет этому пользователю. Как минимум один пользователь должен иметь статус «Включить». Статус «Исключить» можно использовать для исключения определенных пользователей из включенной группы пользователей. Обратите внимание, что пользователи с недостаточными разрешениями (то есть пользователи с ролью на основе типа пользователя *Администратор*, которые не входят в ту же группу пользователей, что и получатель или владелец отчета) вместо фактических имен в полях *Получатель* и *Формировать отчет от имени* будут видеть «Недоступный пользователь» или «Недоступная группа пользователей»; поля *Статус* и *Действие* будут доступны только для чтения.
Включен	Статус отчета. Снятие этого флажка отключит отчет.
Описание	Необязательное описание отчета. Это описание предназначено для внутреннего использования и не будет отправлено получателям отчета.

Кнопки формы

Кнопки в нижней части формы позволяют выполнять несколько операций.

Add	Добавить отчет. Эта кнопка доступна только для новых отчетов.
Update	Обновить свойства отчета.
Clone	Создать другой отчет на основе свойств текущего отчета.
Test	Проверить правильность конфигурации отчета, отправив отчет текущему пользователю.
Delete	Удалить отчет.
Cancel	Отменить редактирование свойств отчета.

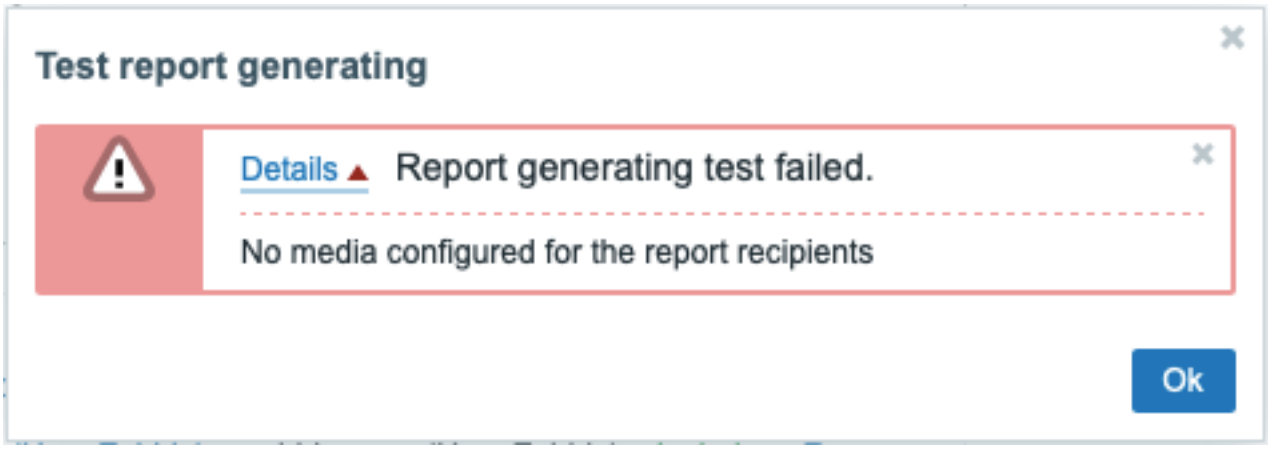
Тестирование

Чтобы протестировать отчет, нажмите кнопку *Тест* в нижней части формы настройки отчета.

Note:
Кнопка *Тест* недоступна, если форма настройки отчета открыта из меню действий панели мониторинга.

Если настройки указаны правильно, тестовый отчет немедленно отправляется текущему пользователю. Для тестовых отчетов настройки подписчиков и пользователя в поле *Создавать отчет от имени* игнорируются.

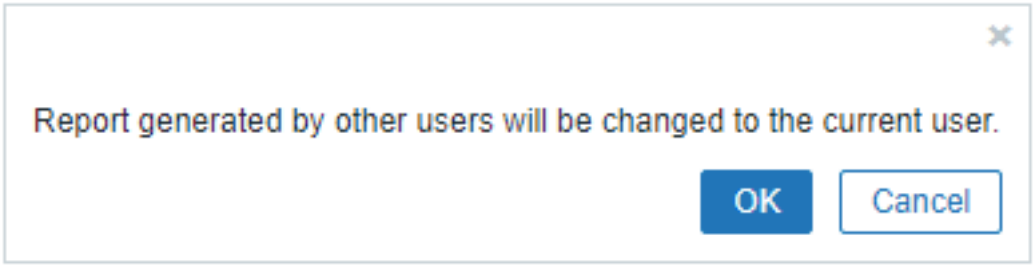
Если настройки указаны неправильно, отображается сообщение об ошибке с описанием возможной причины.



Изменение отчёта

Чтобы изменить существующий отчёт, нажмите на имя отчёта, затем внесите необходимые изменения и нажмите на кнопку *Обновить (Update)*.

Если существующий отчёт обновлён другим пользователем и этот пользователь меняет Панель, при нажатии на кнопку *Обновить* отобразится сообщение с предупреждением: «Отчёт, сгенерированный другими пользователями, изменится на текущего пользователя».



Нажатие *OK* на этом шаге приведёт к следующим изменениям:

- Настройки *Сгенерировано (Generate report by)* будут обновлены, чтобы отображался тот пользователь, который редактировал отчёт в последний раз (если только *Сгенерировано* не задано значением Получатель).
- Пользователи, которые отображались как «Недоступный пользователь» или «Недоступная группа пользователей», будут удалены из списка подписчиков отчёта.

Нажатие *Отмена (Cancel)* закроет всплывающее окно и отменит обновление отчёта.

Клонирование отчёта

Для быстрого клонирования существующего отчёта нажмите кнопку *Клонировать (Clone)* внизу диалога настройки существующего отчёта. При клонировании отчёта, созданного другим пользователем, текущий пользователь становится владельцем нового отчёта.

Настройки отчета будут скопированы в диалог настройки нового отчёта с учётом прав пользователя:

- Если пользователь, который клонирует отчёт, не имеет прав доступа к панели, поле *Панель (Dashboard)* очищается.
- Если пользователь, который клонирует отчёт, не имеет прав доступа к некоторым пользователям или группам пользователей из списка *Подписки (Subscriptions)*, недоступные получатели не клонируются.
- Настройки *Сгенерировано (Generate report by)* будут обновлены, чтобы отображался текущий пользователь (если только *Сгенерировано* не задано значением Получатель).

Измените требуемые настройки и имя отчёта, затем нажмите на *Добавить (Add)*.

Экспорт данных

Обзор

Zabbix поддерживает экспорт данных в режиме реального времени двумя способами:

- экспорт в файлы
- потоковая передача во внешние системы

Могут быть экспортированы следующие объекты:

- триггерные события
- значения элементов данных
- динамика изменений (только экспорт в файлы)

Экспорт в файлы

Обзор

Можно настроить экспорт в реальном времени событий триггеров, значений элементов данных и трендов в формате JSON, где каждый объект отделен новой строкой.

Экспорт выполняется в файлы, где каждая строка файла экспорта является объектом JSON. Преобразования значений не применяются.

В случае ошибок (данные не могут быть записаны в файл экспорта, файл экспорта не может быть переименован или после переименования не может быть создан новый файл), элемент данных отбрасывается и никогда не записывается в файл экспорта. Он записывается только в базу данных Zabbix. Запись данных в файл экспорта возобновляется, когда проблема с записью будет устранена.

Файл экспорта создается с правами на чтение и запись только для владельца файла. Кроме того, файл доступен для чтения группе владельца. Все остальные права запрещены.

Обратите внимание, что узел сети/элемент данных может не иметь метаданных (группы узлов сети, имя узла сети, имя элемента данных), если узел сети/элемент данных был удален после получения данных, но до того, как сервер экспортировал данные.

Настройка

Экспорт событий триггеров, значений элементов данных и трендов в режиме реального времени настраивается путем указания каталога для файлов экспорта - см. параметр ExportDir в конфигурации сервера.

Доступны еще два параметра:

- `ExportFileSize` позволяет задать максимально допустимый размер отдельного файла экспорта. Когда процессу необходимо записать данные в файл, сначала проверяется размер файла. Если он превышает настроенный предел, файл переименовывается путем добавления к его имени `.old`, а затем создается новый файл с исходным именем.

Attention:
Для каждого процесса, который будет записывать данные, создается отдельный файл (то есть примерно 4-30 файлов). Поскольку размер файла экспорта по умолчанию составляет 1G, хранение больших файлов экспорта может быстро заполнить дисковое пространство.

- `ExportType` позволяет указать, какие типы сущностей (события, история, тренды) будут экспортироваться.

Протокол экспорта JSON с разделением по строкам

В этом разделе приведены сведения о протоколе экспорта в формате JSON с разделением по строкам, который используется в:

- экспорте данных в файлы
- потоковой передаче во внешние системы

Все файлы имеют расширение `.ndjson`. Каждая строка файла экспорта представляет собой объект JSON.

Экспорт события триггера

Для события проблемы экспортируется следующая информация:

Field	Type	Description
clock	number	Количество секунд с начала эпохи на момент обнаружения проблемы (целая часть).
ns	number	Количество наносекунд, которое нужно добавить к `clock`, чтобы получить точное время обнаружения проблемы.
value	number	1 (всегда).
eventid	number	ID события проблемы.
name	string	Имя события проблемы.
severity	number	Уровень серьезности события проблемы (0 - Не классифицировано, 1 - Информация, 2 - Предупреждение, 3 - Средняя, 4 - Высокая, 5 - Чрезвычайная).

Field		Type	Description
hosts		array	Список узлов сети, участвующих в выражении триггера; в массиве должен быть как минимум один элемент.
-		object	
	host	string	Имя узла сети.
	name	string	Отображаемое имя узла сети.
groups		array	Список групп узлов сети для всех узлов сети, участвующих в выражении триггера; в массиве должен быть как минимум один элемент.
-		string	Имя группы узлов сети.
tags		array	Список тегов проблемы (может быть пустым).
-		object	
	tag	string	Имя тега.
	value	string	Значение тега (может быть пустым).

Для события восстановления экспортируется следующая информация:

Field	Type	Description
clock	number	Количество секунд с начала эпохи на момент, когда проблема была решена (целая часть).
ns	number	Количество наносекунд, которое нужно добавить к `clock`, чтобы получить точное время устранения проблемы.
value	number	0 (всегда).
eventid	number	ID события восстановления.
p_eventid	number	ID события проблемы.

Примеры экспорта событий триггера

Проблема:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

Восстановление:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Проблема (генерация нескольких событий проблемы):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

Восстановление:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Экспорт значения элемента данных

Следующая информация экспортируется для собранного значения элемента данных:

Field		Type	Description
host		object	Имя узла сети элемента данных.
	host	string	Имя узла сети.
	name	string	Отображаемое имя узла сети.
groups		array	Список групп узлов сети элемента данных; в массиве должен быть как минимум один элемент.
-		string	Имя группы узлов сети.
item_tags		array	Список тегов элемента данных (может быть пустым).
-		object	
	tag	string	Имя тега.
	value	string	Значение тега (может быть пустым).
itemid		number	ID элемента данных.

Field	Type	Description
name	string	Отображаемое имя элемента данных.
clock	number	Количество секунд с начала эпохи на момент, когда было собрано значение (целая часть).
ns	number	Количество наносекунд, которое нужно добавить к `clock`, чтобы получить точное время сбора значения.
timestamp (*Log* only)	number	0, если недоступно.
source (*Log* only)	string	Пустая строка, если недоступно.
severity (*Log* only)	number	0, если недоступно.
eventid (*Log* only)	number	0, если недоступно.
value	number (for numeric items) or string (for text items)	Собранное значение элемента данных.
type	number	Тип собранного значения: 0 - numeric float, 1 - character, 2 - log, 3 - numeric unsigned, 4 - text, 5 - binary, 6 - JSON

Пример экспорта значения элемента данных

Числовое значение (без знака):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Числовое значение (с плавающей запятой):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Символьное, текстовое значение:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Значение журнала:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Экспорт трендов

Следующая информация экспортируется для вычисленного значения тренда:

Field	Type	Description
host	object	Имя узла сети элемента данных.
host	string	Имя узла сети.
name	string	Отображаемое имя узла сети.
groups	array	Список групп узлов сети элемента данных; в массиве должен быть как минимум один элемент.
-	string	Имя группы узлов сети.
item_tags	array	Список тегов элемента данных (может быть пустым).
-	object	
tag	string	Имя тега.
value	string	Значение тега (может быть пустым).
itemid	number	ID элемента данных.
name	string	Отображаемое имя элемента данных.
clock	number	Количество секунд с начала эпохи на момент, когда было собрано значение (целая часть).
count	number	Количество значений, собранных за данный час.
min	number	Минимальное значение элемента данных за данный час.
avg	number	Среднее значение элемента данных за данный час.

Field	Type	Description
max	number	Максимальное значение элемента данных за данный час.
type	number	Тип значения: 0 - numeric float, 3 - numeric unsigned

Примеры экспорта трендов

Числовое значение (без знака):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Числовое значение (float):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Потоковая передача во внешние системы

Обзор

Можно передавать значения элементов данных и события из Zabbix во внешние системы по HTTP (см. описание протокола).

Фильтр по тегам можно использовать для передачи подмножеств значений элементов данных или событий.

За потоковую передачу данных отвечают два типа процессов Zabbix: `connector manager` и `connector worker`. Внутренний элемент данных Zabbix `zabbix[connector_queue]` позволяет отслеживать количество значений, помещенных в очередь прокси-соединителя.

Настройка

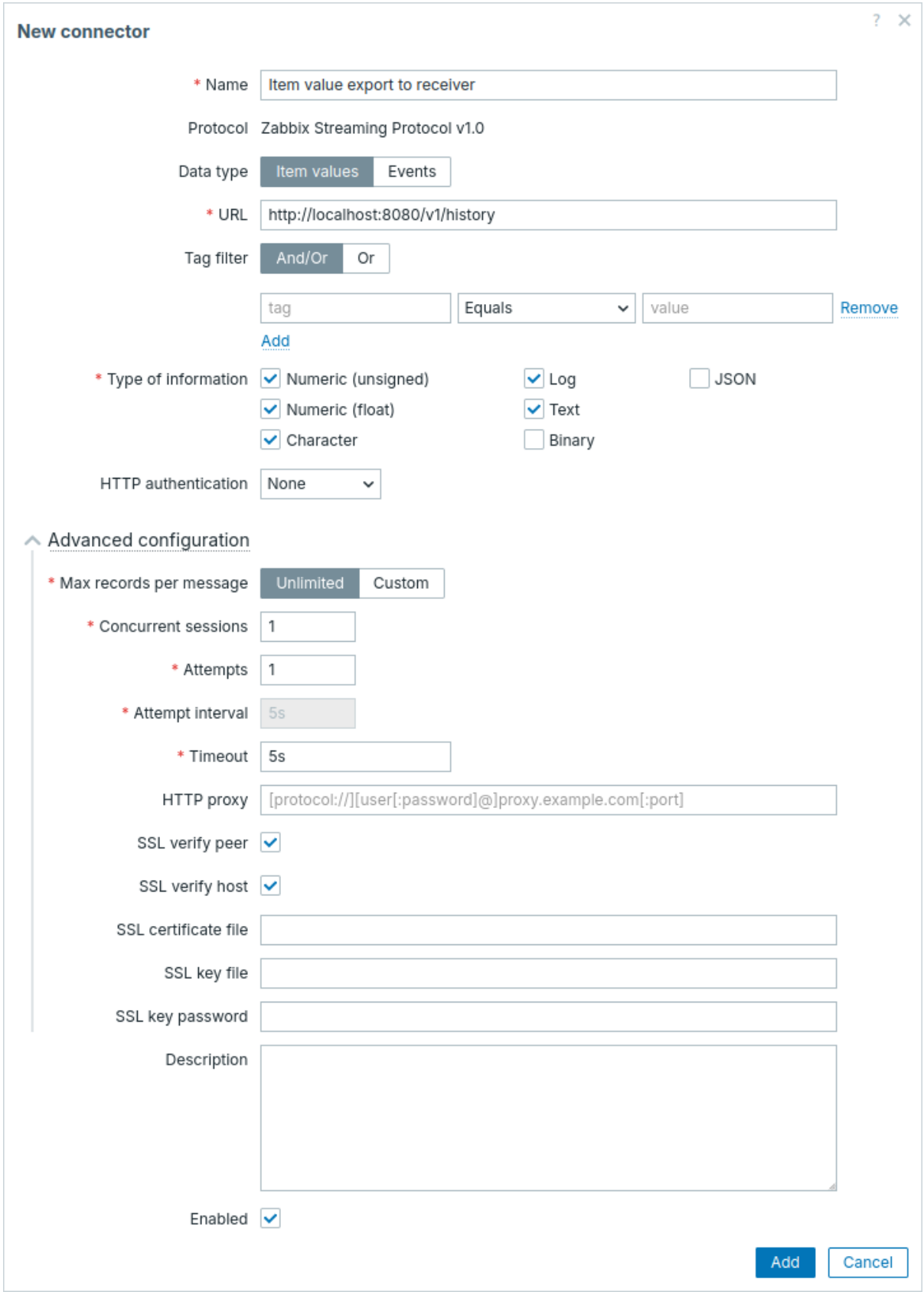
Для настройки потоковой передачи данных во внешнюю систему выполните следующие действия:

1. Настройте удалённую систему для получения данных от Zabbix. Для этого доступны следующие инструменты:

- Пример простого приёмника, который записывает полученную информацию в файлы `events.ndjson` и `history.ndjson`.
- Коннектор Kafka для сервера Zabbix - лёгкий сервер, написанный на Go и предназначенный для передачи значений элементов данных и событий с сервера Zabbix брокеру Kafka.

2. Укажите необходимое количество рабочих процессов коннекторов в Zabbix, изменив параметр `StartConnectors` в `zabbix_server.conf`. Количество рабочих процессов коннекторов должно соответствовать количеству настроенных коннекторов в веб-интерфейсе Zabbix (или превышать его, если количество параллельных сеансов больше 1). Затем перезапустите сервер Zabbix.

3. Настройте новый коннектор в веб-интерфейсе Zabbix (*Администрирование* > *Общие* > *Коннекторы*) и перезагрузите кэш сервера с помощью команды `zabbix_server -R config_cache_reload`.



Обязательные поля отмечены звёздочкой.

Параметр	Описание
Имя	Введите имя коннектора.
Тип данных	Выберите тип данных для потоковой передачи: **Значения элементов данных** - передавать значения элементов данных из Zabbix во внешние системы; **События** - передавать события из Zabbix во внешние системы.
URL	Введите URL приёмника. Поддерживаются пользовательские макросы.

Параметр	Описание
Фильтр тегов	Экспортировать только значения элементов данных или события, соответствующие фильтру тегов. Если фильтр не задан, экспортируются все данные. Можно включать и исключать определённые теги и значения тегов. Можно задать несколько условий. Сопоставление имён тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учётом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введённую строку (поиск подстроки без учёта регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учётом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введённую строку (поиск подстроки без учёта регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега группируются с помощью условия ИЛИ; **ИЛИ** - достаточно выполнения одного условия.
Тип информации	Выберите тип информации (числовой (без знака), числовой (с плавающей запятой), символьный и т. д.), по которому следует фильтровать значения элементов данных, передаваемые коннектором. Это поле доступно, если для параметра *Тип данных* установлено значение «Значения элементов данных».
HTTP-аутентификация	Выберите вариант аутентификации: **Нет** - аутентификация не используется; **Basic** - используется базовая аутентификация; **NTLM** - используется аутентификация NTLM (Windows NT LAN Manager); **Kerberos** - используется аутентификация Kerberos (см. также: Настройка Kerberos с Zabbix); **Digest** - используется дайджест-аутентификация; **Bearer** - используется аутентификация Bearer.
Имя пользователя	Введите имя пользователя (до 255 символов). Поддерживаются пользовательские макросы. Это поле доступно, если для параметра *HTTP-аутентификация* установлено значение «Basic», «NTLM», «Kerberos» или «Digest».
Пароль	Введите пароль пользователя (до 255 символов). Поддерживаются пользовательские макросы. Это поле доступно, если для параметра *HTTP-аутентификация* установлено значение «Basic», «NTLM», «Kerberos» или «Digest».
Токен Bearer	Введите токен Bearer. Поддерживаются пользовательские макросы. Это поле доступно и обязательно, если для параметра *HTTP-аутентификация* установлено значение «Bearer».
Расширенная настройка	Нажмите заголовок *Расширенная настройка*, чтобы отобразить параметры расширенной настройки (см. ниже).
Максимальное количество записей в сообщении	Укажите максимальное количество значений или событий, которые могут передаваться в одном сообщении.
Параллельные сеансы	Выберите количество процессов отправки для запуска этим коннектором. Можно указать до 100 сеансов; значение по умолчанию - «1».
Попытки	Количество попыток потоковой передачи данных. Можно указать до 5 попыток; значение по умолчанию - «1».
Интервал между попытками	Укажите, как долго коннектор должен ожидать после неудачной попытки передачи данных. Можно указать до 10 с; значение по умолчанию - «5 с». Это поле доступно, если для параметра *Попытки* установлено значение «2» или больше. Неудачными считаются попытки, при которых не удалось установить соединение или код ответа HTTP не равен 200, 201, 202, 203 или 204. Повторные попытки **запускаются** в случае ошибок связи или если код ответа HTTP не равен 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415 или 422. Перенаправления обрабатываются, поэтому 302 -> 200 является положительным ответом, а 302 -> 503 приведёт к повторной попытке.

Параметр	Описание
Тайм-аут	Укажите тайм-аут сообщения (1-60 секунд, значение по умолчанию - 5 секунд). Поддерживаются суффиксы времени (например, 30s, 1m). Поддерживаются пользовательские макросы.
HTTP-прокси	Можно указать HTTP-прокси в следующем формате: `[protocol://][username[:password]@]proxy.example.com[:port]` Поддерживаются пользовательские макросы. Необязательный префикс `protocol://` можно использовать для указания альтернативных протоколов прокси (поддержка префикса протокола добавлена в cURL 7.21.7). Если протокол не указан, прокси будет считаться HTTP-прокси. По умолчанию используется порт 1080. Если указан параметр *HTTP-прокси*, прокси переопределит переменные окружения, связанные с прокси, например `http_proxy`, `HTTPS_PROXY`. Если параметр не указан, прокси не будет переопределять переменные окружения, связанные с прокси. Введённое значение передаётся без изменений, проверка корректности не выполняется. Также можно ввести адрес SOCKS-прокси. Если указать неправильный протокол, коннектор не сможет передавать значения элементов данных или события из Zabbix. Обратите внимание, что при использовании HTTP-прокси поддерживается только простая аутентификация.
Проверять SSL-сертификат узла сети	Установите флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера проверяется по сертификатам доверенных центров сертификации (CA) из общесистемного расположения CA. Расположение файлов CA можно переопределить с помощью параметра конфигурации сервера Zabbix или прокси `SSLCALocation`.
Проверять SSL-узел сети	Установите флажок, чтобы проверять соответствие поля *Common Name* или *Subject Alternate Name* сертификата веб-сервера. При этом устанавливается параметр cURL `CURLOPT_SSL_VERIFYHOST`.
Файл SSL-сертификата	Имя файла SSL-сертификата, используемого для аутентификации клиента. Файл сертификата должен быть в формате PEM[1]. Поддерживаются пользовательские макросы. Если файл сертификата также содержит закрытый ключ, оставьте поле *Файл SSL-ключа* пустым. Если ключ зашифрован, укажите пароль в поле *Пароль SSL-ключа*. Каталог, содержащий этот файл, задаётся параметром конфигурации сервера Zabbix или прокси `SSLCertLocation`.
Файл SSL-ключа	Имя файла закрытого SSL-ключа, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM[1]. Поддерживаются пользовательские макросы. Каталог, содержащий этот файл, задаётся параметром конфигурации сервера Zabbix или прокси `SSLKeyLocation`.
Пароль SSL-ключа	Пароль файла закрытого SSL-ключа. Поддерживаются пользовательские макросы.
Описание	Введите описание коннектора.
Включён	Установите флажок, чтобы включить коннектор.

Если коннектор Kafka настроен с разделённым запятыми списком адресов начальных брокеров (например, `Kafka.URL=kafka1.exam`
клиент Kafka подключается к брокеру или брокерам, которые отвечают первыми, и использует метаданные их кластера. Если список содержит адреса из разных кластеров Kafka, будет использоваться только кластер, ответивший быстрее всего, а остальные адреса будут записаны в журнал как недоступные; в результате могут появиться предупреждения при запуске, подобные следующему, даже если коннектор подключён:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

В некоторых средах (частных сетях, сетях контейнеров или конфигурациях с нестандартными настройками DNS/hosts) имена узлов сети или IP-адреса могут разрешаться в loopback-адреса (например, `127.0.0.1`/`localhost`) или нормализоваться клиентом, из-за чего такие предупреждения могут вводить в заблуждение. Чтобы уменьшить вероятность путаницы, убедитесь, что все адреса `Kafka.URL` принадлежат одному кластеру Kafka, проверьте разрешение DNS с узла сети коннектора и параметры `advertised.listeners` брокеров, а также отдавайте предпочтение адресам, которые разрешаются в рекламируемый адрес брокера.

Протокол

Обмен между сервером и приемником осуществляется по HTTP с использованием REST API, NDJSON, "Content-Type:

application/x-ndjson”.

Подробнее см. Протокол экспорта JSON, разделенного переводами строк.

Запрос сервера

Пример потоковой передачи значений элементов данных:

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

Пример потоковой передачи событий:

```
POST /v1/events HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 333
Content-Type: application/x-ndjson

{"clock":1673454303,"ns":800155804,"value":1,"eventid":5,"name":"trigger for foo being 0","severity":0,"ho
{"clock":1673454303,"ns":832290669,"value":0,"eventid":6,"p_eventid":5}
```

Ответ получателя

Ответ состоит из кода состояния ответа HTTP и полезной нагрузки JSON. Код состояния ответа HTTP должен быть «200», «201», «202», «203» или «204» для запросов, которые были обработаны успешно, и другим для невыполненных запросов.

Пример успешного приёма:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Пример с ошибками:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

SNMP gateway

Обзор

Zabbix SNMP gateway — это AgentX-расширение для snmpd, поддерживающее как SNMP-опрос, так и SNMP-trap.

С помощью Zabbix SNMP gateway можно использовать протокол SNMP для получения:

- данных триггера;
- данных проблемного триггера;
- состояния группы узлов сети (количество триггеров по состоянию триггера в каждой группе)

Данные извлекаются по OID, который представляет собой комбинацию общего базового значения и определенного суффикса. Общее **базовое значение** задается в конфигурационном файле SNMP gateway, например:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - для любых данных триггера;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - для данных проблемного триггера;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - для состояния группы узлов сети.

Суффикс OID задается в конфигурации триггеров узла сети как **тег** (например, `OIDSuffix:3`) в веб-интерфейсе.

В этом случае вся информация о триггере будет доступна по `OID=1.3.6.1.4.1.3043.7.55.X.3`. Здесь "X" — это номер поля данных триггера (то есть 1 — суффикс, 2 — ID, 3 — выражение, 4 — описание и т. д.).

Более подробное описание и пример конфигурационного файла см. в readme-файле SNMP gateway.

Установка и настройка

См. файл readme шлюза SNMP для инструкций по:

- установке и настройке snmpd;
- включению поддержки AgentX;
- настройке шлюза SNMP Zabbix;
- настройке SNMP traps для изменений состояния триггера.

Получение данных

При правильной настройке вы можете использовать команды `snmpwalk` и `snmpget` для получения данных:

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

Параметры фильтрации

Вы можете ограничить информацию о триггерах с проблемами в конфигурации SNMP gateway:

- по уровню серьезности (по умолчанию `ProblemMinSeverity=-1`)
- скрывая подтвержденные проблемы (по умолчанию `ProblemHideAck=false`)

Вы можете ограничить количество проблем на группу узлов сети в конфигурации SNMP gateway:

- по триггерам в неизвестном состоянии (по умолчанию `CountUnknown=false`)
- по триггерам с подтвержденными/неподтвержденными/всеми проблемами (по умолчанию `CountAcknowledgeStatus=all`)

Экспорт/импорт конфигурации

Обзор

Функциональность экспорта/импорта Zabbix позволяет обмениваться различными объектами конфигурации между одной системой Zabbix и другой.

Типичные варианты использования этой функциональности:

- совместное использование шаблонов или сетевых карт - пользователи Zabbix могут делиться своими параметрами конфигурации
- загрузить шаблон в Zabbix Community templates. Затем другие пользователи могут скачать шаблон и импортировать файл в Zabbix.

- интеграция со сторонними инструментами - универсальные форматы YAML, XML и JSON делают возможной интеграцию и импорт/экспорт данных со сторонними инструментами и приложениями

Что можно экспортировать/импортировать

Объекты, которые можно экспортировать/импортировать:

- Группы узлов сети (*только через Zabbix API*)
- Группы шаблонов (*только через Zabbix API*)
- Шаблоны
- Узлы сети
- Карты сети
- Типы медиа
- Изображения
- Панели
- Регулярные выражения

Формат экспорта

Данные можно экспортировать с помощью веб-интерфейса Zabbix или Zabbix API. Поддерживаемые форматы экспорта: YAML, XML и JSON.

Подробности об экспорте

- Все поддерживаемые элементы экспортируются в один файл.
- Сущности узла сети и шаблона (элементы данных, триггеры, графики, правила обнаружения), унаследованные от связанных шаблонов, не экспортируются. Любые изменения, внесенные в эти сущности на уровне узла сети (например, измененный интервал элемента данных, измененное регулярное выражение или добавленные прототипы к правилу низкоуровневого обнаружения), будут потеряны при экспорте; при импорте все сущности из связанных шаблонов воссоздаются так же, как в исходном связанном шаблоне.
- Сущности, созданные низкоуровневым обнаружением, и любые сущности, зависящие от них, не экспортируются. Например, триггер, созданный для элемента данных, сгенерированного правилом LLD, не будет экспортирован.
- Когда экспортируемый узел сети/шаблон содержит сущности, поддерживающие тайм-ауты, значения тайм-аутов будут экспортированы, если для этих сущностей настроены собственные тайм-ауты.

Сведения об импорте

- Импорт останавливается при первой ошибке.
- При обновлении существующих изображений во время импорта изображений поле ”imagetype” игнорируется, то есть изменить тип изображения через импорт невозможно.
- При импорте узлов сети/шаблонов с использованием опции ”Удалить отсутствующие” макросы узла сети/шаблона, отсутствующие в файле импорта, будут удалены из узла сети/шаблона после импорта.
- Пустые теги для элементов данных, триггеров, графиков, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes не имеют смысла, то есть это то же самое, как если бы они отсутствовали.
- Если для сущностей импортируемого узла сети/шаблона настроены собственные тайм-ауты, будут применены они; в противном случае будут применены тайм-ауты прокси/глобальные тайм-ауты.
- Импорт поддерживает YAML, XML и JSON, файл импорта должен иметь правильное расширение: .yaml и .yml для YAML, .xml для XML и .json для JSON. См. сведения о совместимости о поддерживаемых версиях XML.
- Импорт поддерживает только файлы конфигурации в кодировке UTF-8 (с BOM или без него); другие кодировки (UTF16LE, UTF16BE, UTF32LE, UTF32BE и т. д.) приведут к ошибке преобразования импорта.

Базовый формат YAML

Формат экспорта YAML содержит следующие узлы:

- Корневой узел для экспорта Zabbix в YAML
- Версия экспорта

```
zabbix_export:
  version: '8.0'
```

Другие узлы зависят от экспортируемых объектов.

XML-формат

Формат экспорта XML содержит следующие теги:

- Стандартный заголовок для XML-документов
- Корневой тег для XML-экспорта Zabbix
- Версия экспорта

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

Другие теги зависят от экспортируемых объектов.

Формат JSON

Формат экспорта JSON содержит следующие объекты:

- Корневой объект для экспорта Zabbix JSON
- Версия экспорта

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Другие объекты зависят от экспортируемых объектов.

Группы шаблонов

Обзор

Во веб-интерфейсе группы шаблонов можно экспортировать только вместе с экспортом шаблона. При экспорте шаблона все группы, к которым он принадлежит, автоматически экспортируются вместе с ним.

API позволяет экспортировать группы шаблонов независимо от шаблонов.

Формат экспорта

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: '           '
```

Экспортированные элементы

Element	Type	Description
uuid	string	Уникальный идентификатор этой группы шаблонов.
name	string	Имя группы.

Группы узлов сети

Обзор

В веб-интерфейсе группы узлов сети можно экспортировать только вместе с экспортом узла сети. При экспорте узла сети все группы, к которым он принадлежит, автоматически экспортируются вместе с ним.

API позволяет экспортировать группы узлов сети независимо от узлов сети.

Формат экспорта

```
   host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: '     Zabbix'
```

Экспортированные элементы

Element	Type	Description
uuid	string	Уникальный идентификатор этой группы узлов сети.
name	string	Имя группы.

Шаблоны

Обзор

Шаблоны экспортируются вместе со многими связанными объектами и связями между объектами.

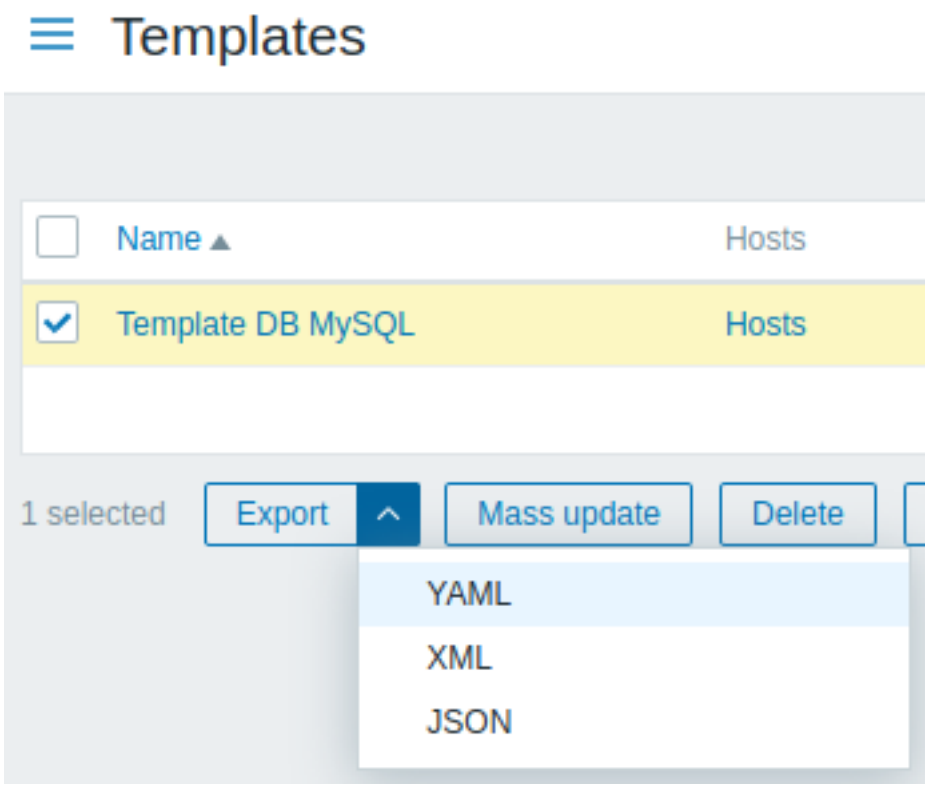
Экспорт шаблона содержит:

- Связанные группы шаблонов
- Связанные группы узлов сети (если используются в конфигурации прототипа узла сети)
- Данные шаблона
- Связи с другими шаблонами
- Связи с группами шаблонов
- Непосредственно связанные элементы данных
- Непосредственно связанные триггеры
- Непосредственно связанные графики
- Непосредственно связанные панели мониторинга
- Непосредственно связанные правила обнаружения со всеми прототипами
- Непосредственно связанные веб-сценарии
- Карты значений

Экспорт

Чтобы экспортировать шаблоны, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Шаблоны*.
2. Отметьте флажками шаблоны, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



В зависимости от выбранного формата шаблоны экспортируются в локальный файл с именем по умолчанию:

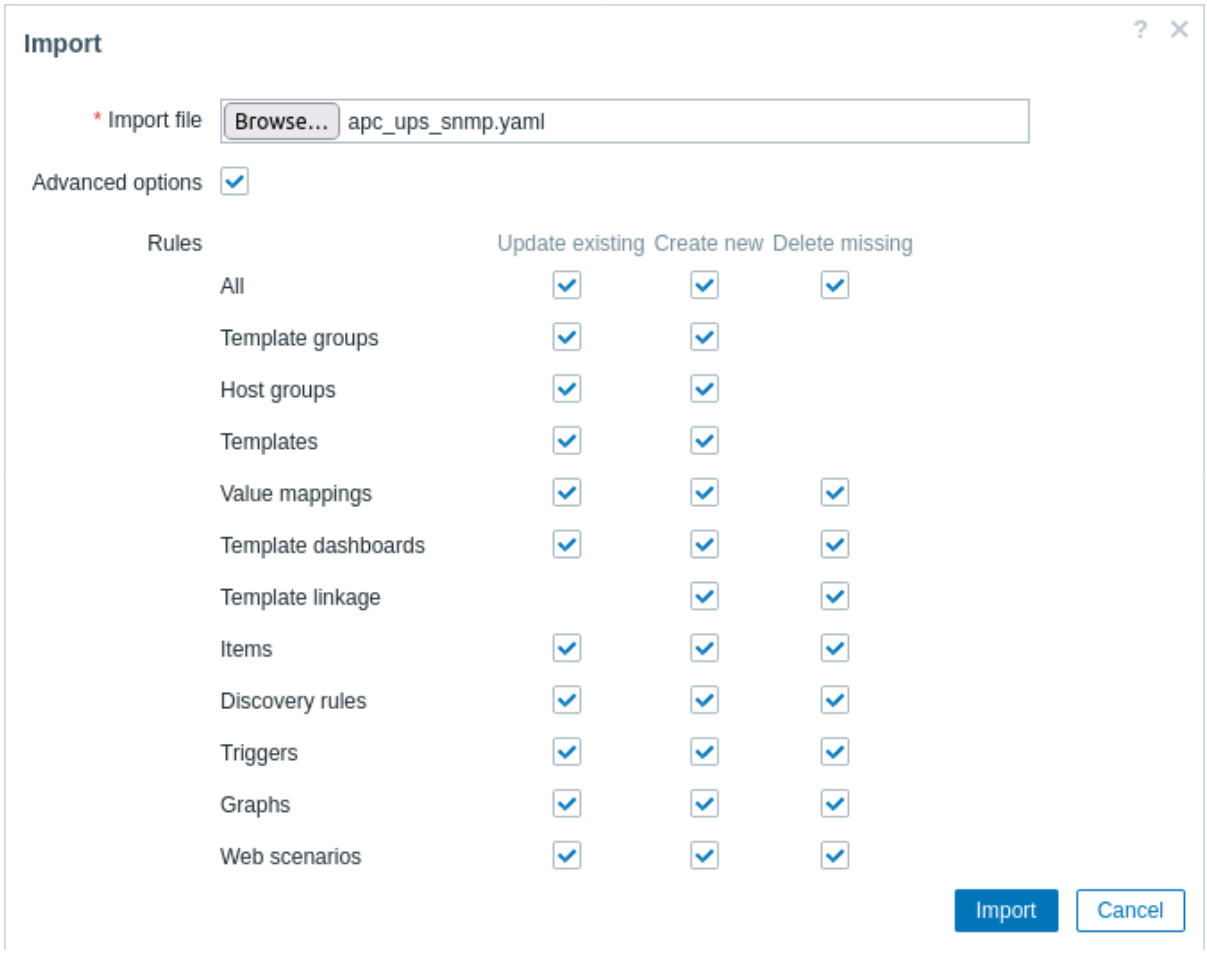
- `zabbix_export_templates.yaml` - при экспорте в YAML;
- `zabbix_export_templates.xml` - при экспорте в XML;
- `zabbix_export_templates.json` - при экспорте в JSON.

Импорт

Чтобы импортировать шаблоны, выполните следующие действия:

1. Перейдите в раздел *Сбор данных* > *Шаблоны*.
2. Нажмите *Импорт* в правом верхнем углу.

3. Выберите файл импорта.
4. Установите флажки правил импорта, которые хотите применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Если установить флажок *Расширенные параметры*, отобразится подробный список всех доступных для импорта элементов. Установите или снимите флажки правил импорта по необходимости.

Если установить флажок в строке *Все*, будут установлены или сняты флажки для всех расположенных ниже элементов.

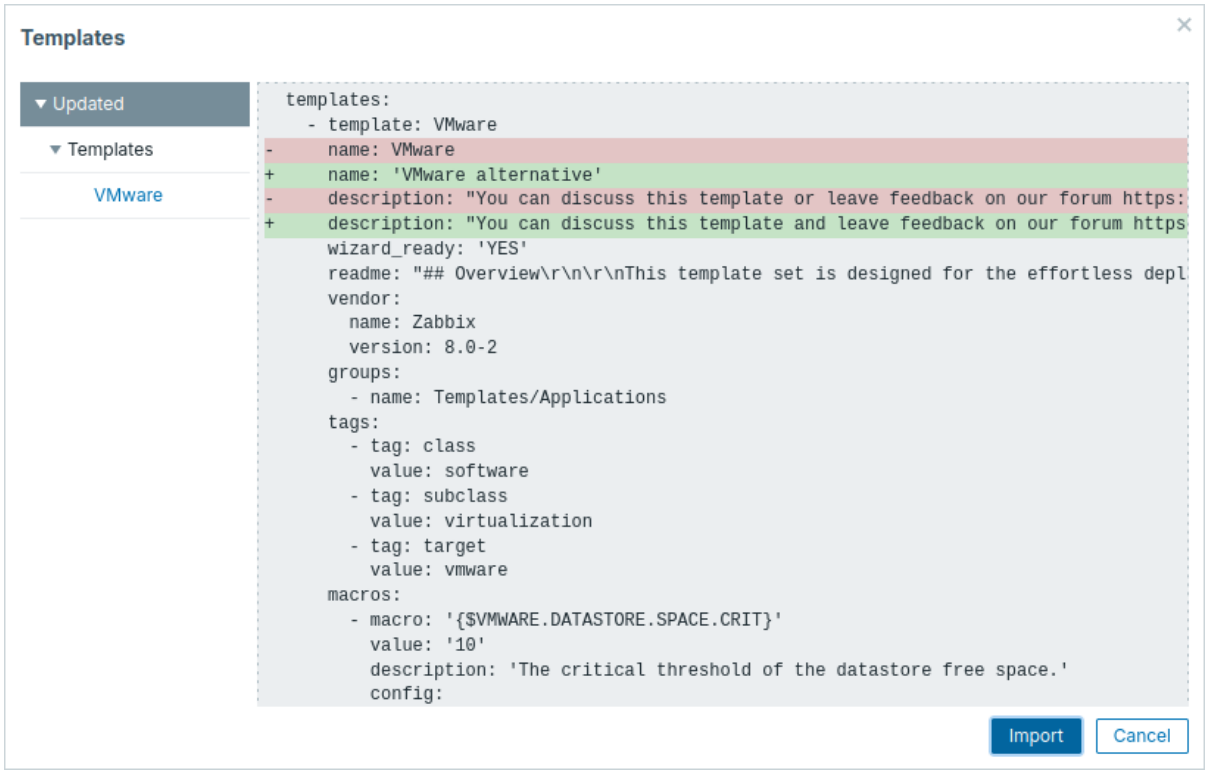
Правила импорта:

Правило	Описание
Обновлять существующие	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создавать новые	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.
Удалять отсутствующие	Существующие элементы, отсутствующие в файле импорта, будут удалены. В противном случае они не будут удалены. Если для *Связей шаблона* установлен флажок *Удалять отсутствующие*, текущие связи шаблона, отсутствующие в файле импорта, будут удалены. Сущности (элементы данных, триггеры, графики и т. д.), унаследованные от отвязанных шаблонов, удалены не будут (если только для каждой сущности также не выбран параметр *Удалять отсутствующие*).

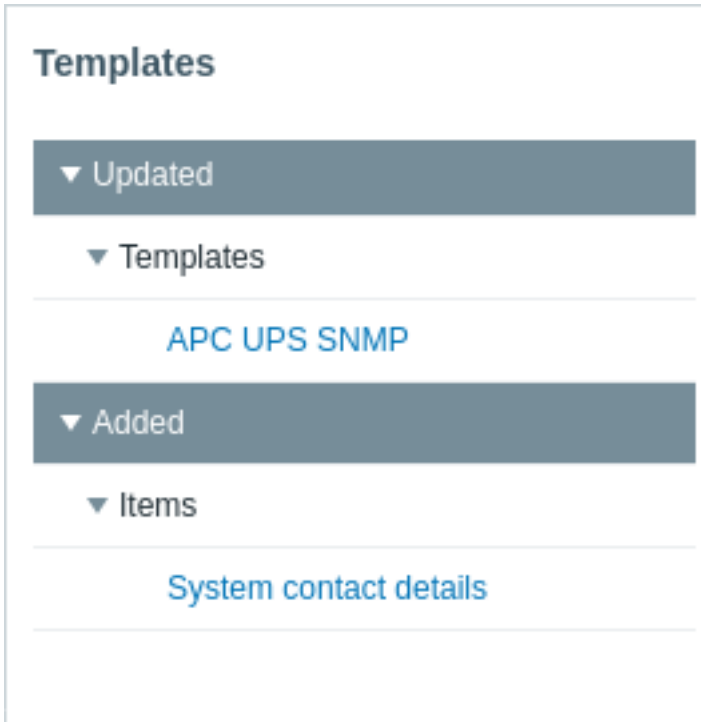
Attention:
Если шаблоны с такими же именами уже существуют, при импорте следует установить параметры *Удалять отсутствующие*, чтобы выполнить чистый импорт. Таким образом будут удалены старые элементы данных, которых больше нет в обновленном шаблоне (обратите внимание, что это приведет к потере истории этих старых элементов данных).

На следующем экране можно просмотреть содержимое импортируемого шаблона. Если это новый шаблон, все элементы будут выделены зеленым цветом. При обновлении существующего шаблона новые элементы шаблона будут выделены

зеленым цветом, удаленные элементы шаблона — красным, а неизменившиеся элементы будут отображаться на сером фоне.



Меню слева можно использовать для навигации по списку изменений. В разделе *Обновлено* выделены все изменения, внесенные в существующие элементы шаблона. В разделе *Добавлено* перечислены новые элементы шаблона. Элементы в каждом разделе сгруппированы по типу элемента; нажмите серую стрелку, чтобы развернуть или свернуть группу элементов.



Проверьте изменения шаблона, затем нажмите *Импорт*, чтобы выполнить импорт шаблона. В веб-интерфейсе отобразится сообщение об успешном или неудачном импорте.

Формат экспорта Когда шаблон экспортируется, Zabbix генерирует структурированный формат YAML, JSON или XML. Экспорт включает элементы шаблона, такие как метаданные шаблона, элементы данных, макросы, триггеры, панели мониторинга и многое другое.

Каждый элемент выполняет определенную функцию и может содержать вложенные элементы.

В следующих разделах описывается каждый элемент в формате экспорта. В примерах используется шаблон Linux by Zabbix agent.

Многоточие (...) обозначает элементы, опущенные для краткости. Примечание (.) используется, когда элемент объясняется более подробно в последующих разделах.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: ( .        )
  triggers: ( .       )
  graphs: ( .        )
```

Метаданные шаблона

Element	Type	Description
version	string	(обязательно) Версия экспорта.
template_groups		(обязательно) Корневой элемент для групп шаблонов.
uuid	string	(обязательно) Уникальный идентификатор этой группы шаблонов.
name	string	(обязательно) Имя группы шаблонов.
host_groups		Корневой элемент для групп узлов сети, которые используются прототипами узлов сети.
uuid	string	(обязательно) Уникальный идентификатор этой группы узлов сети.
name	string	(обязательно) Имя группы узлов сети.
templates		Корневой элемент для шаблонов.
triggers		Корневой элемент для элементов триггеров шаблона, которые совпадают с триггерами элементов данных шаблона.
graphs		Корневой элемент для элементов графиков шаблона, которые совпадают с графиками узлов сети.

Шаблоны

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
                        Linux.                Zabbix 8.0       . (...)
      wizard_ready: 'YES'
      readme: |
        ##

                        Linux.                Zabbix 8.0       . (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
```

```
        description: '  -  ,                          .                ,            /    Zabbix (
        config: (see table below)
      - macro: '{$CPU.UTIL.CRIT}'
        value: '90'
        description: '               CPU,         %.'
        config: (see table below)
      (...)
    dashboards: (see table below)
    valuemaps: (see table below)
  (...)
```

Element	Type	Description
uuid	string	(required) Уникальный идентификатор для этого шаблона.
template	string	(required) Уникальное имя шаблона.
name	string	Отображаемое имя шаблона.
description	text	Описание шаблона.
wizard_ready	text	Доступен ли шаблон для выбора в Мастере узлов сети. Возможные значения:[1] NO (0, по умолчанию), YES (1). См. также: Объект Template (`wizard_ready`).
readme	text	Инструкции по настройке, специфичные для шаблона, которые отображаются в Мастере узлов сети. Поддерживает форматирование Markdown.
vendor		Корневой элемент для поставщика шаблона (присутствует, если экспортированный шаблон содержит данные о поставщике).
name	string	(required) Имя поставщика шаблона.
version	string	(required) Версия шаблона. Для готовых шаблонов версия отображается следующим образом: основная версия Zabbix, разделитель (”-”), номер ревизии (увеличивается с каждой новой версией шаблона и сбрасывается с каждой основной версией Zabbix). Например, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Корневой элемент для связанных шаблонов.
name	string	(required) Имя шаблона.
groups		Корневой элемент для групп шаблонов.
name	string	(required) Имя группы шаблонов.
items		Корневой элемент для элементов данных шаблона.
discovery_rules		Корневой элемент для правил низкоуровневого обнаружения шаблона.
httptests		Корневой элемент для веб-сценариев шаблона.
tags		Корневой элемент для тегов шаблона.
tag	string	(required) Имя тега.
value	string	Значение тега.
macros		Корневой элемент для пользовательских макросов шаблона.
macro	string	(required) Имя пользовательского макроса.
type	string	Тип пользовательского макроса. Возможные значения:[1] TEXT (0, по умолчанию), SECRET_TEXT (1), VAULT (2). См. также: Объект User macro (`type`).
value	string	Значение пользовательского макроса.
description	string	Описание пользовательского макроса.
config		Корневой элемент для настройки макроса шаблона, отвечающий за то, как макрос отображается в Мастере узлов сети.
dashboards		Корневой элемент для панелей мониторинга шаблона.
valuemaps		Корневой элемент для карт значений шаблона.

Элементы данных шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
    items:
      - uuid: f94f9f4699e94c369e6c98b2a2f485ce
        name: '                Zabbix'
        key: agent.ping
        description: '              «1»              .                  `nodata()`              .'
```

```
      valuemap:
        name: '                    Zabbix'
      tags:
        - tag: component
          value: system
    (...)
    - uuid: 58818005e76d46dda14d6592f601ab00
      name: '                   '
      key: system.sw.packages.get
      delay: 1h
      preprocessing: ( .        )
      tags:
        - tag: component
          value: os
      triggers: ( .        )
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: '            '
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: '                             `100-pavailable`.'
      preprocessing: ( .        )
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: ( .        )
    (...)
```

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этого элемента данных.
name	string	(обязательно) Имя элемента данных.
type	string	Тип элемента данных. Возможные значения:[1] ZABBIX_PASSIVE (0, по умолчанию), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), TELEMETRY_QUERY (24).
snmp_oid	string	(обязательно для элементов данных SNMP_AGENT) Идентификатор объекта SNMP.
key	string	(обязательно) Ключ элемента данных.
delay	string	Интервал обновления элемента данных. По умолчанию: 1m. Для элементов данных TRAP значение всегда равно 0.
history	string	Период времени (с использованием суффикса времени, макроса пользователя или макроса LLD), в течение которого должны храниться данные истории. По умолчанию: 31d.
trends	string	Период времени (с использованием суффикса времени, макроса пользователя или макроса LLD), в течение которого должны храниться данные трендов. По умолчанию: 365d.
status	string	Состояние элемента данных. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).
value_type	string	Тип полученного значения. Возможные значения:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, по умолчанию), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Список IP-адресов узлов сети, разделённых запятыми, которым разрешено отправлять данные для элемента данных. Поддерживается для элементов данных TRAP и HTTP_AGENT.
units	string	Единицы измерения полученного значения (bps, B и т. д.).

Элемент	Тип	Описание
params	text	Дополнительные параметры в зависимости от типа элемента данных (исполняемый скрипт для элементов данных SSH и TELNET; SQL-запрос для элементов данных ODBC; формула для элементов данных CALCULATED; скрипт для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER).
ipmi_sensor	string	Датчик IPMI. Поддерживается для элементов данных IPMI.
authtype	string	Тип аутентификации. Поддерживается для элементов данных SSH и HTTP_AGENT. Возможные значения для элементов данных SSH:[1] PASSWORD (0, по умолчанию), PUBLIC_KEY (1). Возможные значения для элементов данных HTTP_AGENT:[1] NONE (0, по умолчанию), BASIC (1), NTLM (2), Kerberos (3) или Digest (4).
username	string	(обязательно для элементов данных SSH и TELNET) Имя пользователя для аутентификации. Поддерживается для элементов данных SIMPLE, ODBC, JMX и HTTP_AGENT. При использовании для элементов данных JMX также следует указать `password` (см. ниже) либо оставить оба элемента пустыми.
password	string	(обязательно для элементов данных SSH и TELNET) Пароль для аутентификации. Поддерживается для элементов данных SIMPLE, ODBC, JMX и HTTP_AGENT. При использовании для элементов данных JMX также следует указать `username` (см. выше) либо оставить оба элемента пустыми.
publickey	string	(обязательно для элементов данных SSH) Имя файла открытого ключа.
privatekey	string	(обязательно для элементов данных SSH) Имя файла закрытого ключа.
description	text	Описание элемента данных.
inventory_link	string	Поле инвентаризации узла сети, которое заполняется элементом данных. Возможные значения:[1] NONE (0), ALIAS (4) и т. д. (поддерживаемые поля см. в разделе Инвентаризация узла сети).
valuemap		Корневой элемент для отображений значений элемента данных.
name	string	(обязательно) Имя отображения значений, используемого для элемента данных.
logtimefmt	string	Формат времени в записях журнала. Поддерживается для элементов данных с типом значения LOG.
preprocessing		Корневой элемент для предварительной обработки значений элемента данных.
step		Корневой элемент для шагов предварительной обработки значений элементов данных шаблона.
jmx_endpoint	string	Конечная точка JMX. Поддерживается для элементов данных JMX.
master_item		(обязательно для зависимых элементов данных) Корневой элемент главного элемента данных зависимого элемента данных.
key	string	(обязательно) Ключ главного элемента данных зависимого элемента данных.
timeout	string	Тайм-аут запроса на получение данных элемента данных. Поддерживается для типов элементов данных из списка Тайм-ауты.
url	string	(обязательно для элементов данных HTTP_AGENT) Строка URL.
query_fields		Корневой элемент для параметров запроса. Поддерживается для элементов данных HTTP_AGENT.
name	string	(обязательно для элементов данных HTTP_AGENT) Имя параметра запроса.
value	string	Значение параметра запроса. Поддерживается для элементов данных HTTP_AGENT.
parameters		Корневой элемент для пользовательских параметров. Поддерживается для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER.
name	string	(обязательно для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER) Имя пользовательского параметра.
value	string	Значение пользовательского параметра. Поддерживается для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER.
posts	string	Данные тела запроса HTTP(S). Поддерживается для элементов данных HTTP_AGENT.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Поддерживается для элементов данных HTTP_AGENT.
follow_redirects	string	Следовать перенаправлениям ответа при получении данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0), YES (1, по умолчанию).

Элемент	Тип	Описание
post_type	string	Тип тела данных POST. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (2), XML (3).
http_proxy	string	Строка подключения к прокси HTTP(S). Поддерживается для элементов данных HTTP_AGENT.
headers		Корневой элемент для заголовков запроса HTTP(S). Поддерживается для элементов данных HTTP_AGENT.
name	string	(обязательно для элементов данных HTTP_AGENT) Имя заголовка.
value	string	(обязательно для элементов данных HTTP_AGENT) Значение заголовка.
retrieve_mode	string	Какая часть ответа должна сохраняться. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] BODY (0, по умолчанию), HEADERS (1), BOTH (2).
request_method	string	Тип метода запроса. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] GET (0, по умолчанию), POST (1), PUT (2), HEAD (3).
output_format	string	Способ обработки ответа. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (1).
allow_traps	string	Разрешить заполнение значения аналогично элементу данных trapper. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
ssl_key_password	string	Пароль для файла ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
verify_peer	string	Проверять ли подлинность сертификата узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
verify_host	string	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
time_shift	string	Насколько далеко в прошлое (до текущего времени) заканчивается временной диапазон, запрашиваемый элементом данных. Поддерживается для элементов данных TELEMETRY_QUERY.
lookback_limit	string	Насколько далеко в прошлое (до текущего времени) элемент данных запрашивает данные, если отсутствует предыдущий запрос, от которого можно продолжить (например, при первом запуске элемента данных), или если предыдущий запрос старше `delay` (например, из-за недоступности сервера Zabbix). Поддерживается для элементов данных TELEMETRY_QUERY.
granularity	string	Длительность фрагмента, на который разбивается полный временной диапазон запроса. Поддерживается для элементов данных TELEMETRY_QUERY.
query		(обязательно для элементов данных TELEMETRY_QUERY) Корневой элемент запроса телеметрии. Определяет, что элемент данных запрашивает из источника данных, как он фильтрует данные и как агрегирует результат.
signal_type	string	(обязательно) Тип сигнала, запрашиваемого элементом данных. Поддерживается для элементов данных TELEMETRY_QUERY. Возможные значения:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(обязательно для типа сигнала METRICS) Тип точки метрики, запрашиваемой элементом данных. Поддерживается для элементов данных TELEMETRY_QUERY. Возможные значения:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).

Элемент	Тип	Описание
columns		(обязательно) Корневой элемент для столбцов данных телеметрии, запрашиваемых элементом данных. Этот элемент содержит следующие элементы: `column` (`string`) - (обязательно) Имя столбца данных телеметрии, запрашиваемого элементом данных. `attribute_key` (`string`) - (обязательно для столбцов, содержащих пары ключ-значение) Имя ключа, который запрашивает элемент данных, в столбце, содержащем пары ключ-значение.
aggregated_columns		(обязательно) Корневой элемент для агрегированных столбцов данных телеметрии, запрашиваемых элементом данных. Этот элемент содержит следующие элементы: `column` (`string`) - (обязательно для функций MIN, MAX, AVG, SUM и PERCENTILE) Имя столбца данных телеметрии, запрашиваемого элементом данных. `function` (`string`) - (обязательно) Функция, применяемая элементом данных к записям столбца. Возможные значения:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (обязательно для функций PERCENTILE) Процентиль, вычисляемый элементом данных по записям столбца. `alias` (`string`) - (обязательно) Уникальное имя вычисленного значения, используемое для его идентификации в значении элемента данных.
filter		(обязательно) Корневой элемент фильтра запроса телеметрии для записей столбцов в запросе. Этот элемент содержит следующие элементы: `evaltype` (`string`) - (обязательно) Метод вычисления условий фильтра. Возможные значения:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (обязательно для методов вычисления FORMULA) Определяемое пользователем выражение для вычисления условий фильтров с пользовательским выражением. `conditions` (`string`) - (обязательно) Корневой элемент условий фильтра данных телеметрии, которым должна соответствовать запись в столбце данных телеметрии. Элемент `conditions` содержит следующие элементы: `column` (`string`) - (обязательно) Имя столбца данных телеметрии, к которому применяются условия фильтра. `attribute_key` (`string`) - (обязательно для столбцов, содержащих пары ключ-значение) Имя ключа, к которому применяется условие, в столбце, содержащем пары ключ-значение. `operator` (`string`) - (обязательно) Оператор условия фильтра. Возможные значения для столбцов с одиночными значениями:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Возможные значения для столбцов, содержащих пары ключ-значение:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (обязательно для методов вычисления FORMULA) Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. `value` (`string`) - (обязательно для операций EQUAL, NOT_EQUAL, LIKE и NOT LIKE) Значение для сравнения.
tags		Корневой элемент тегов элемента данных.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
triggers		Корневой элемент для триггеров элементов данных шаблона.

Note:
См. также: Объект элемента данных (соответствующее свойство с совпадающим именем).

Шаги предварительной обработки значения элемента данных шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Description
type	string	(обязательно) Тип шага предварительной обработки значения элемента данных. Возможные значения:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, вычисляется: полученное значение - предыдущее значение), CHANGE_PER_SECOND (10, вычисляется: (полученное значение - предыдущее значение)/(текущее время - время последней проверки)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(обязательно) Корневой элемент для параметров шага предварительной обработки значения элемента данных.
parameter	string	Отдельный параметр шага предварительной обработки значения элемента данных.
error_handler	string	Тип действия, используемый в случае сбоя шага предварительной обработки. Возможные значения:[1] ORIGINAL_ERROR (0, по умолчанию), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Параметры обработчика ошибок.

Note:
См. также: Объект предварительной обработки элемента данных (см. соответствующее свойство с совпадающим именем).

Триггеры элемента данных шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Number of installed packages has been changed'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
```

```
                value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: High memory utilization'
              event_name: 'Linux: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)'
              priority: AVERAGE
              description: 'The system is running out of free memory.'
              dependencies:
                - name: 'Linux: Lack of available memory'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

Element	Type	Description
uuid	string	(required) Уникальный идентификатор этого триггера.
expression	string	(required) Выражение триггера.
recovery_mode	string	Основа для генерации событий OK. Возможные значения:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Выражение восстановления триггера.
correlation_mode	string	Режим корреляции (без корреляции событий или корреляция событий по тегу). Возможные значения:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	Имя тега, используемое для корреляции событий.
name	string	(required) Имя триггера.
event_name	string	Имя события.
opdata	string	Оперативные данные.
url_name	string	Метка для URL, связанного с триггером.
url	string	URL, связанный с триггером.
status	string	Статус триггера. Возможные значения:[1] ENABLED (0, default), DISABLED (1).
priority	string	Серьезность триггера. Возможные значения:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Описание триггера.
type	string	Тип генерации событий (одно проблемное событие или несколько проблемных событий). Возможные значения:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	Ручное закрытие проблемных событий. Возможные значения:[1] NO (0, default), YES (1).
dependencies		Корневой элемент для зависимостей.
name	string	(required) Имя зависимого триггера.
expression	string	(required) Выражение зависимого триггера.
recovery_expression	string	Выражение восстановления зависимого триггера.
tags		Корневой элемент для тегов триггера.
tag	string	(required) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект Trigger (см. соответствующее свойство с совпадающим именем).

Правила низкоуровневого обнаружения шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: '                    .'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
Большинство элементов правила низкоуровневого обнаружения шаблона совпадают с элементами элементов данных шаблона. В таблице ниже описаны только те элементы, которые отличаются от элементов данных шаблона.

Element	Type	Description
type	string	Тип элемента данных. Возможные значения:[1] ZABBIX_PASSIVE (0, default), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(required) Ключ правила низкоуровневого обнаружения.
filter		Корневой элемент для фильтров правил низкоуровневого обнаружения шаблона.
lifetime	string	Период времени (в секундах, суффикс времени или пользовательский макрос), по истечении которого обнаруженные ресурсы, которые больше не обнаруживаются, будут удалены. По умолчанию: 7d.
lifetime_type	string	Сценарий удаления потерянных ресурсов LLD. Возможные значения: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Период времени (в секундах, суффикс времени или пользовательский макрос), по истечении которого обнаруженные ресурсы, которые больше не обнаруживаются, будут отключены.
enabled_lifetime_type	string	Сценарий отключения потерянных ресурсов LLD. Возможные значения: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Корневой элемент для элементов-прототипов шаблона, которые совпадают с элементами данных шаблона.
trigger_prototypes		Корневой элемент для триггеров-прототипов шаблона, которые совпадают с триггерами элементов данных шаблона.
graph_prototypes		Корневой элемент для графиков-прототипов шаблона, которые совпадают с графиками узла сети.
host_prototypes		Корневой элемент для прототипов узлов сети шаблона, которые совпадают с узлами сети.
parent_discovery_rule		Корневой элемент для родительского правила низкоуровневого обнаружения (или прототипа правила) прототипа правила низкоуровневого обнаружения. Это свойство указывает, что это прототип правила LLD, прямой дочерний элемент указанного правила/прототипа правила.
key	string	(required) Ключ родительского правила низкоуровневого обнаружения (или прототипа правила).
master_item	string	(required for DEPENDENT rules) Корневой элемент для master-элемента зависимого правила.
lld_macro_paths		Корневой элемент для путей макросов правила низкоуровневого обнаружения.
lld_macro	string	(required) Имя макроса правила низкоуровневого обнаружения.
path	string	(required) Селектор значения, которое будет назначено соответствующему макросу.
preprocessing		Корневой элемент для предварительной обработки значения правила низкоуровневого обнаружения.

Element	Type	Description
step		Корневой элемент для элементов шага предварительной обработки значения правила низкоуровневого обнаружения, которые совпадают с шагами предварительной обработки значения элемента данных шаблона, но с меньшим количеством возможных значений. См. также: объект предварительной обработки правила LLD.
overrides		Корневой элемент для правил переопределения правила низкоуровневого обнаружения.
name	string	(required) Уникальное имя переопределения.
step	string	(required) Уникальный порядковый номер переопределения.
stop	string	Остановить обработку следующих переопределений, если совпадает.
filter		Корневой элемент для элементов фильтра правила переопределения правила низкоуровневого обнаружения шаблона, которые совпадают с фильтрами правил низкоуровневого обнаружения шаблона.
operations		Корневой элемент для операций переопределения правила низкоуровневого обнаружения шаблона.

Note:
См. также: объект правила LLD (ссылайтесь на соответствующее свойство с совпадающим именем).

Фильтры правила низкоуровневого обнаружения шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
        - uuid: 2bbdc79f082d4c618e01bec625e9c90a
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.MATCHES}'
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
              - macro: '{#DEVTYPE}'
                value: disk
          (...)
```

Element	Type	Description
evaltype	string	Переопределяет метод оценки условий фильтра. Возможные значения:[1] AND_OR (0, по умолчанию), AND (1), OR (2), FORMULA (3).
formula	string	Пользовательская формула вычисления для условий фильтра.
conditions		Корневой элемент для условий фильтра.
macro	string	(required) Имя макроса низкоуровневого обнаружения, для которого выполняется проверка.
value	string	Значение для сравнения.

Element	Type	Description
operator	string	Оператор условия. Возможные значения:[1] MATCHES_REGEX (8, по умолчанию), NOT_MATCHES_REGEX (9).
formulaid	string	(required) Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть задан пользователем при изменении условий фильтра, но при последующем запросе будет сгенерирован заново.

Note:
См. также: объект фильтра правила LLD (см. соответствующее свойство с совпадающим именем).

Операции переопределения правила низкоуровневого обнаружения шаблона

Element	Type	Description
operationobject	string	Объект, к которому применяется операция. Возможные значения:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Оператор условия переопределения. Возможные значения:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Регулярное выражение или строка для оператора условия переопределения.
status	string	Статус объекта после операции переопределения.
discover	string	Добавляется ли объект в результате обнаружения.
delay	string	Интервал обновления, заданный для прототипа элемента данных после операции переопределения.
history	string	Период хранения истории, заданный для прототипа элемента данных после операции переопределения.
trends	string	Период хранения трендов, заданный для прототипа элемента данных после операции переопределения.
severity	string	Уровень серьезности прототипа триггера, заданный после операции переопределения.
tags		Корневой элемент для набора тегов, заданных для объекта после операции переопределения.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
templates		Корневой элемент для шаблонов, связанных с прототипом узла сети после операции переопределения.
name	string	(обязательно) Имя шаблона.
inventory_mode	string	Режим инвентаризации прототипа узла сети, заданный после операции переопределения.

Note:
См. также: Объект операции переопределения правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Веб-сценарии шаблона

Element	Type	Description
uuid	string	(required) Уникальный идентификатор для этого веб-сценария.
name	string	(required) Имя веб-сценария.
delay	string	Частота выполнения веб-сценария (в секундах, суффикс времени или пользовательский макрос). По умолчанию: 1m.
attempts	integer	Количество попыток выполнения шагов веб-сценария. Допустимые значения:[1] 1-10 (по умолчанию: 1).

Element	Type	Description
agent	string	Клиентский агент. Zabbix будет выдавать себя за выбранный браузер. Это полезно, когда веб-сайт возвращает разное содержимое для разных браузеров. По умолчанию: Zabbix.
http_proxy	string	Прокси, который будет использоваться веб-сценарием, задается в виде: `http://[username[:password]@]proxy.example.com[:port]`
variables		Корневой элемент для переменных веб-сценария, которые могут использоваться в шагах сценария.
name	string	(required) Имя переменной.
value	text	(required) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, которые будут отправляться при выполнении запроса. Заголовки следует перечислять, используя тот же синтаксис, что и в HTTP-протоколе.
name	string	(required) Имя заголовка.
value	text	(required) Значение заголовка.
status	string	Статус веб-сценария. Допустимые значения:[1] ENABLED (0, default), DISABLED (1).
authentication	string	Метод аутентификации. Допустимые значения:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3) или Digest (4).
http_user	string	Имя пользователя, используемое для аутентификации BASIC (HTTP), NTLM, Kerberos или Digest.
http_password	string	Пароль, используемый для аутентификации BASIC (HTTP), NTLM, Kerberos или Digest.
verify_peer	string	Проверять SSL-сертификат веб-сервера. Допустимые значения:[1] NO (0, default), YES (1).
verify_host	string	Проверять, что поле *Common Name* или поле *Subject Alternate Name* сертификата веб-сервера совпадает. Допустимые значения:[1] NO (0, default), YES (1).
ssl_cert_file	string	Имя файла SSL-сертификата, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_file	string	Имя файла закрытого ключа SSL, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль файла закрытого ключа SSL.
steps		(required) Корневой элемент для шагов веб-сценария шаблона.
tags		Корневой элемент для тегов веб-сценария.
tag	string	(required) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект веб-сценария (обратитесь к соответствующему свойству с совпадающим именем).

Шаги веб-сценария шаблона

Element	Type	Description
name	string	(required) Имя шага веб-сценария.
url	string	(required) URL для мониторинга.
query_fields		Корневой элемент для параметров запроса (массив HTTP-полей, которые будут добавлены к URL при выполнении запроса).
name	string	(required) Имя параметра запроса.
value	string	Значение параметра запроса.
posts		Корневой элемент для переменных HTTP POST (строка (raw post data) или массив HTTP-полей (данные полей формы)).
name	string	(required) Имя поля POST.
value	string	(required) Значение поля POST.

Element	Type	Description
variables		Корневой элемент переменных уровня шага (макросов), которые должны быть применены после этого шага. Если значение переменной имеет префикс 'regex:', то его значение извлекается из данных, возвращенных этим шагом, в соответствии с шаблоном регулярного выражения, следующим за префиксом 'regex:'
name	string	(required) Имя переменной.
value	text	(required) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, которые будут отправлены при выполнении запроса.
name	string	(required) Имя заголовка.
value	text	(required) Значение заголовка.
follow_redirects	string	Следовать HTTP-перенаправлениям. Возможные значения:[1] NO (0), YES (1, по умолчанию).
retrieve_mode	string	Режим получения HTTP-ответа. Возможные значения:[1] BODY (0, по умолчанию), HEADERS (1), BOTH (2).
timeout	string	Тайм-аут (в секундах, time suffix или user macro) выполнения шага. По умолчанию: 15s.
required	string	Текст, который должен присутствовать в ответе (игнорируется, если пусто).
status_codes	string	Список допустимых HTTP-кодов состояния, разделенных запятыми (например, 200-201,210-299; игнорируется, если пусто).

Note:
См. также: Объект шага веб-сценария (см. соответствующее свойство с совпадающим именем).

Конфигурация макроса шаблона

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: '                              Zabbix'
            description: '   -  ,                            .'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: '                CPU'
            description: '                        CPU   %.         0    100         .'
            regex: '^-?([0-9]+|([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: '             '
            description: '          ,                                "Link down".'
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Элемент	Тип	Описание
type	string	(обязательно) Тип поля ввода макроса. Возможные значения:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Позиция макроса в списке макросов.
section_name	string	Метка сворачиваемого раздела, в котором сгруппирован макрос.
label	string	(обязательно для макросов TEXT, LIST, CHECKBOX) Метка макроса.
description	text	Справочный текст макроса. Поддерживает форматирование Markdown.
required	string	Помечает макрос как обязательный. Возможные значения:[1] NO (0), YES (1). Поддерживается для макросов TEXT и LIST.
regex	string	Регулярное выражение для проверки пользовательского ввода в поле текстового ввода. Поддерживается для макросов TEXT.
options		Корневой элемент для элементов LIST или значений CHECKBOX.
value	string	(обязательно для макросов LIST) Значение элемента LIST.
text	string	(обязательно для макросов LIST) Текст элемента LIST.
checked	string	(обязательно для макросов CHECKBOX) Значение, представляющее состояние "включено". Возможные значения:[1] 0 (false), 1 (true). Поддерживается для макросов CHECKBOX.
unchecked	string	(обязательно для макросов CHECKBOX) Значение, представляющее состояние "выключено". Возможные значения:[1] 0 (false), 1 (true). Поддерживается для макросов CHECKBOX.

Note:
См. также: Объект конфигурации макроса (см. соответствующее свойство с совпадающим именем).

Панели шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этой панели.
name	string	(обязательно) Имя панели шаблона.
display pe-riod	integer	Период отображения страниц панели.
auto_start	string	Автоматический запуск слайд-шоу. Возможные значения:[1] NO (0), YES (1, по умолчанию).
pages		Корневой элемент для страниц панели шаблона.
name	string	Имя страницы.
display period	integer	Интервал слайд-шоу.
sortorder	integer	Порядок сортировки страницы.
widgets		Корневой элемент для виджетов панели шаблона.

Note:
См. также: Объект панели шаблона (см. соответствующее свойство с совпадающим именем).

Виджеты панели шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Description
type	string	(обязательно) Тип виджета.
name	string	Имя виджета.
x	integer	Горизонтальная позиция от левой стороны панели шаблона. Возможные значения:[1] 0-71.
y	integer	Вертикальная позиция от верхней стороны панели шаблона. Возможные значения:[1] 0-63.
width	integer	Ширина виджета. Возможные значения:[1] 1-72.
height	integer	Высота виджета. Возможные значения:[1] 1-64.
hide_header	string	Скрыть заголовок виджета. Возможные значения:[1] NO (0, по умолчанию), YES (1).
fields		Корневой элемент для полей виджета панели шаблона.
type	string	(обязательно) Тип поля виджета. Возможные значения:[1] INTEGER (0), STRING (1), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).

Element	Type	Description
name	string	(обязательно) Имя поля виджета.
value	mixed	(обязательно) Значение поля виджета, в зависимости от типа поля.

Note:
См. также: Объект виджета панели шаблона (см. соответствующее свойство с совпадающим именем).

Карта значений шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
              newvalue: up
        (...)
```

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этой карты значений.
name	string	(обязательно) Имя карты значений.
mapping		Корневой элемент для сопоставлений.
type	string	Тип сопоставления по совпадению. Возможные значения:[1] EQUAL (0, по умолчанию), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Исходное значение.
newvalue	string	(обязательно) Значение, в которое сопоставляется исходное значение.

Note:
См. также: Объект карты значений (см. соответствующее свойство с совпадающим именем).

Сноски [1] Целочисленные значения API в скобках, например ENABLED (0), приведены только для справки. Для получения дополнительной информации см. связанную страницу объекта API в записи таблицы или в конце каждого раздела.

Узлы сети

Обзор

Узлы сети экспортируются вместе со многими связанными объектами и связями между объектами.

Экспорт узла сети содержит:

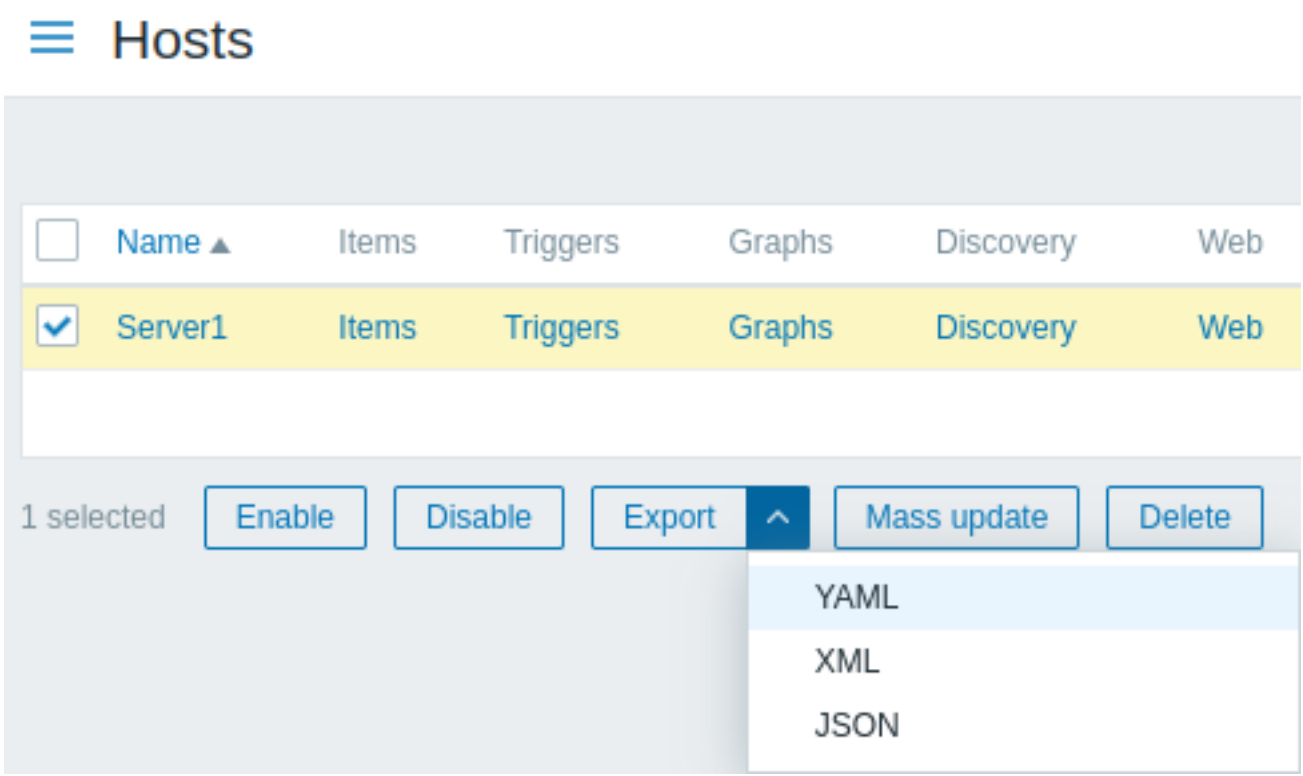
- Связанные группы узлов сети
- Данные узла сети
- Связи с шаблонами

- Связи с группами узлов сети
- Интерфейсы узла сети
- Непосредственно связанные элементы данных
- Непосредственно связанные триггеры
- Непосредственно связанные правила обнаружения со всеми прототипами
- Непосредственно связанные веб-сценарии
- Макросы узла сети
- Данные инвентаризации узла сети
- Карты значений
- Связанные графики

Экспорт

Чтобы экспортировать узлы сети, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Отметьте флажками узлы сети, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



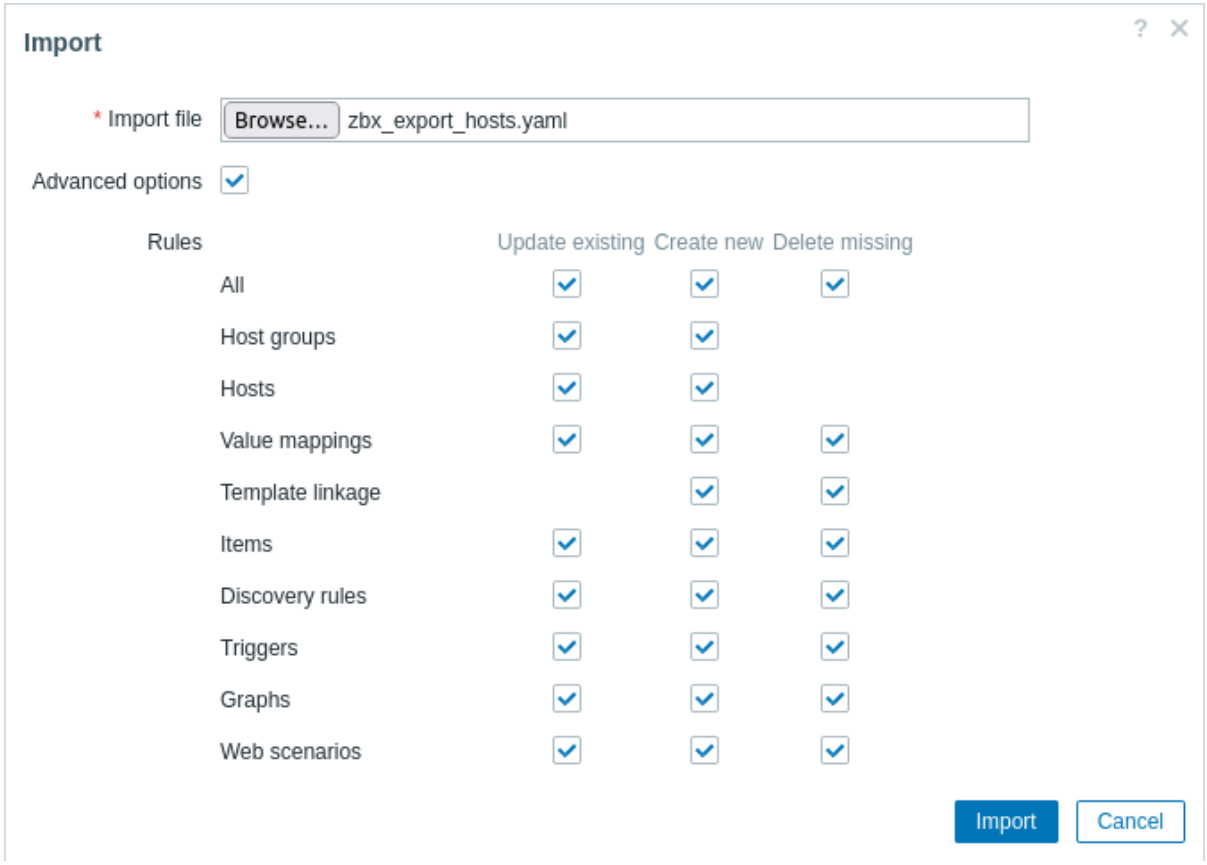
В зависимости от выбранного формата узлы сети экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_hosts.yaml` - при экспорте в YAML;
- `zabbix_export_hosts.xml` - при экспорте в XML;
- `zabbix_export_hosts.json` - при экспорте в JSON.

Импортирование

Чтобы импортировать узлы сети, выполните следующие действия:

1. Перейдите в раздел *Сбор данных* > *Узлы сети*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл импорта.
4. Установите флажки правил импорта, которые необходимо применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Если установить флажок *Расширенные параметры*, отобразится подробный список всех доступных для импорта элементов - установите или снимите флажок каждого правила импорта по необходимости.

Если установить флажок в строке *Все*, будут установлены или сняты флажки всех расположенных ниже элементов.

Правила импорта:

Правило	Описание
Обновлять существующие	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создавать новые	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.
Удалять отсутствующие	Существующие элементы, отсутствующие в файле импорта, будут удалены. В противном случае они не будут удалены. Если для *Связи с шаблонами* установлено правило *Удалять отсутствующие*, текущие связи с шаблонами, отсутствующие в файле импорта, будут удалены. Сущности (элементы данных, триггеры, графики и т. д.), унаследованные от отключенных шаблонов, удалены не будут (если только для каждой сущности также не выбран параметр *Удалять отсутствующие*).

Во время импорта невозможно просмотреть изменения узлов сети. Сообщение об успешном или неудачном импорте будет отображено в веб-интерфейсе.

Формат экспорта

Когда узел сети экспортируется, Zabbix генерирует структурированный формат YAML, JSON или XML. Экспорт включает элементы узла сети, такие как метаданные узла сети, элементы данных, макросы, триггеры и многое другое.

Каждый элемент служит определенной цели и может содержать вложенные элементы.

В следующих разделах описан каждый элемент в формате экспорта. В примерах используется узел сети с шаблоном MySQL by Zabbix agent 2. Для демонстрации элементов шаблон отвязывается после создания узла сети. При отвязке все элементы остаются в конфигурации (в отличие от отвязки и очистки, которая удаляет их).

Многоточие (`...`) обозначает элементы, опущенные для краткости. Примечание (.) используется, когда элемент более подробно объясняется в последующих разделах.

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: ( .        )
  graphs: ( .        )
```

Метаданные узла сети

Element	Type	Description
version	string	(обязательно) Версия Zabbix.
host_groups		(обязательно) Корневой элемент для групп узлов сети.
uuid	string	(обязательно) Уникальный идентификатор для этой группы узлов сети.
name	string	(обязательно) Имя группы узлов сети.
hosts		Корневой элемент для узлов сети.
graphs		Корневой элемент для графиков узла сети.

Узлы сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: '            MySQL.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(required) Уникальное имя узла сети.
name	string	Отображаемое имя узла сети.
description	text	Описание узла сети.
monitored_by	string	Способ мониторинга узла сети. Possible values:[1] SERVER (0, default), PROXY (1), or PROXY_GROUP (2).
proxy		Корневой элемент для прокси.
name	string	(required) Имя прокси (если есть), который мониторит узел сети.
proxy_group		Корневой элемент для группы прокси.
name	string	(required) Имя группы прокси (если есть), которая используется для мониторинга узла сети.
status	string	Статус узла сети. Possible values:[1] ENABLED (0, default), DISABLED (1).

Element	Type	Description
ipmi_authtype	string	Тип аутентификации сеанса IPMI. Possible values:[1] DEFAULT (-1, default), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).
ipmi_privilege	string	Уровень привилегий сеанса IPMI. Possible values:[1] CALLBACK (1), USER (2, default), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Имя пользователя для проверок IPMI.
ipmi_password	string	Пароль для проверок IPMI.
templates		Корневой элемент для связанных шаблонов.
name	string	(required) Имя шаблона.
groups		Корневой элемент для групп узлов сети, к которым принадлежит узел сети.
name	string	(required) Имя группы узлов сети.
interfaces		Корневой элемент для интерфейсов узла сети.
items		Корневой элемент для элементов данных узла сети.
discovery_rules		Корневой элемент для правил низкоуровневого обнаружения узла сети.
http_tests		Корневой элемент для веб-сценариев узла сети.
tags		Корневой элемент для тегов узла сети.
tag	string	(required) Имя тега.
value	string	Значение тега.
macros		Корневой элемент для пользовательских макросов узла сети.
macro	string	(required) Имя пользовательского макроса.
type	string	Тип пользовательского макроса. Possible values:[1] TEXT (0, default), SECRET_TEXT (1), VAULT (2).
value	string	Значение пользовательского макроса.
description	string	Описание пользовательского макроса.
inventory		Корневой элемент для инвентаризации узла сети.
<inventory_property>	string	Свойство инвентаризации. У всех свойств есть соответствующий элемент (type, name, os и т. д.; например, см. Формат экспорта).
inventory_mode	string	Режим инвентаризации. Possible values:[1] DISABLED (-1), MANUAL (0, default), AUTOMATIC (1).
valuemaps		Корневой элемент для карт значений узла сети.

Note:
См. также: Объект Host (обратитесь к соответствующему свойству с совпадающим именем).

Интерфейсы узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Description
default	string	Является ли это основным интерфейсом узла сети. Обратите внимание, что на узле сети может быть только один основной интерфейс одного типа. Возможные значения:[1] NO (0), YES (1, default).
type	string	Тип интерфейса. Возможные значения:[1] ZABBIX (1, default), SNMP (2), IPMI (3), JMX (4).
useip	string	Использовать ли IP в качестве интерфейса для подключения к узлу сети (в противном случае будет использоваться DNS). Возможные значения:[1] NO (0), YES (1, default).
ip	string	(required for IP connections) IP address (IPv4 or IPv6).
dns	string	(required for DNS connections) DNS name.
port	string	Номер порта.
details		Корневой элемент для подробных сведений об интерфейсе.

Element	Type	Description
version	string	Использовать эту версию SNMP. Возможные значения:[1] SNMPV1 (1), SNMP_V2C (2, default), SNMP_V3 (3).
community	string	(required for SNMPv1 and SNMPv2 items) SNMP community.
max_repetitions	string	Максимальное значение повторений для нативных SNMP bulk-запросов (GetBulkRequest-PDUs). Поддерживается для элементов данных SNMPv2 и SNMPv3 (`discovery[]` и `walk[]`). По умолчанию: 10.
contextname	string	Имя контекста SNMPv3. Поддерживается для элементов данных SNMPv3.
securityname	string	Имя безопасности SNMPv3. Поддерживается для элементов данных SNMPv3.
securitylevel	string	Уровень безопасности SNMPv3. Поддерживается для элементов данных SNMPv3. Возможные значения:[1] NOAUTHNOPRIV (0, default), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Протокол аутентификации SNMPv3. Поддерживается для элементов данных SNMPv3. Возможные значения:[1] MD5 (0, default), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Фраза-пароль аутентификации SNMPv3. Поддерживается для элементов данных SNMPv3.
privprotocol	string	Протокол конфиденциальности SNMPv3. Поддерживается для элементов данных SNMPv3. Возможные значения:[1] DES (0, default), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Фраза-пароль конфиденциальности SNMPv3. Поддерживается для элементов данных SNMPv3.
bulk	string	Использовать bulk-запросы для SNMP. Возможные значения:[1] NO (0), YES (1, default).
interface_ref	string	Имя ссылки на интерфейс, которое будет использоваться в элементах данных (формат: `if<N>`).

Note:
См. также: Объект интерфейса узла сети (см. соответствующее свойство с совпадающим именем).

Элементы данных узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: '                          '
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: '            ,                      ,
          preprocessing: ( .        )
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: '                '
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
```

```
        params: |
          ( last(//mysql.innodb_buffer_pool_pages_total) -
          last(//mysql.innodb_buffer_pool_pages_free) ) /
          ( last(//mysql.innodb_buffer_pool_pages_total) +
          ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
          ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
        description: '                                          .'
        tags:
          - tag: component
            value: memory
        triggers: ( .        )
      (...)
      - name:
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: '          ,                    .'
        preprocessing: ( .         )
        master_item:
          key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
        tags:
          - tag: component
            value: application
        triggers: ( .        )
      (...)
```

Элемент	Тип	Описание
name	string	(обязательно) Имя элемента данных.
type	string	Тип элемента данных. Возможные значения:[1] ZABBIX_PASSIVE (0, по умолчанию), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(обязательно для элементов данных SNMP_AGENT) Идентификатор объекта SNMP.
key	string	(обязательно) Ключ элемента данных.
delay	string	Интервал обновления элемента данных. По умолчанию: 1m. Для элементов данных TRAP значение всегда равно 0.
history	string	Период времени (с использованием суффикса времени, пользовательского макроса или макроса LLD), в течение которого должны храниться данные истории. По умолчанию: 31d.
trends	string	Период времени (с использованием суффикса времени, пользовательского макроса или макроса LLD), в течение которого должны храниться данные трендов. По умолчанию: 365d.
status	string	Состояние элемента данных. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).
value_type	string	Тип полученного значения. Возможные значения:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, по умолчанию), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Список IP-адресов узлов сети, разделённых запятыми, которым разрешено отправлять данные для элемента данных. Поддерживается для элементов данных TRAP и HTTP_AGENT.
units	string	Единицы измерения полученного значения (bps, B и т. д.).
params	text	Дополнительные параметры в зависимости от типа элемента данных (исполняемый скрипт для элементов данных SSH и TELNET; SQL-запрос для элементов данных ODBC; формула для элементов данных CALCULATED; скрипт для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER).
ipmi_sensor	string	Датчик IPMI. Поддерживается для элементов данных IPMI.

Элемент	Тип	Описание
authtype	string	Тип аутентификации. Поддерживается для элементов данных SSH и HTTP_AGENT. Возможные значения для элементов данных SSH:[1] PASSWORD (0, по умолчанию), PUBLIC_KEY (1). Возможные значения для элементов данных HTTP_AGENT:[1] NONE (0, по умолчанию), BASIC (1), NTLM (2), Kerberos (3) или Digest (4).
username	string	(обязательно для элементов данных SSH и TELNET) Имя пользователя для аутентификации. Поддерживается для элементов данных SIMPLE, ODBC, JMX и HTTP_AGENT. При использовании для элементов данных JMX также следует указать `password` (см. ниже) либо оставить оба элемента пустыми.
password	string	(обязательно для элементов данных SSH и TELNET) Пароль для аутентификации. Поддерживается для элементов данных SIMPLE, ODBC, JMX и HTTP_AGENT. При использовании для элементов данных JMX также следует указать `username` (см. выше) либо оставить оба элемента пустыми.
publickey	string	(обязательно для элементов данных SSH) Имя файла открытого ключа.
privatekey	string	(обязательно для элементов данных SSH) Имя файла закрытого ключа.
description	text	Описание элемента данных.
inventory_link	string	Поле инвентаризации узла сети, которое заполняется элементом данных. Возможные значения:[1] NONE (0), ALIAS (4) и т. д. (поддерживаемые поля см. в разделе Инвентаризация узла сети).
valuemap		Корневой элемент для сопоставлений значений элементов данных.
name	string	(обязательно) Имя сопоставления значений, используемого для элемента данных.
logtimefmt	string	Формат времени в записях журнала. Поддерживается для элементов данных с типом значения LOG.
preprocessing		Корневой элемент для предварительной обработки значений элементов данных.
step		Корневой элемент для шагов предварительной обработки значений элементов данных узла сети.
interface_ref	string	Ссылка на интерфейс узла сети (формат: `if<N>`).
jmx_endpoint	string	Конечная точка JMX. Поддерживается для элементов данных JMX.
master_item		(обязательно для зависимых элементов данных) Корневой элемент главного элемента данных зависимого элемента данных.
key	string	(обязательно) Ключ главного элемента данных зависимого элемента данных.
timeout	string	Тайм-аут запроса на получение данных элемента данных. Поддерживается для списка Тайм-ауты типов элементов данных.
url	string	(обязательно для элементов данных HTTP_AGENT) Строка URL.
query_fields		Корневой элемент для параметров запроса. Поддерживается для элементов данных HTTP_AGENT.
name	string	(обязательно для элементов данных HTTP_AGENT) Имя параметра запроса.
value	string	Значение параметра запроса. Поддерживается для элементов данных HTTP_AGENT.
parameters		Корневой элемент для пользовательских параметров. Поддерживается для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER.
name	string	(обязательно для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER) Имя пользовательского параметра.
value	string	Значение пользовательского параметра. Поддерживается для элементов данных ITEM_TYPE_SCRIPT и ITEM_TYPE_BROWSER.
posts	string	Данные тела запроса HTTP(S). Поддерживается для элементов данных HTTP_AGENT.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Поддерживается для элементов данных HTTP_AGENT.
follow_redirects	string	Следовать перенаправлениям ответа при получении данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0), YES (1, по умолчанию).
post_type	string	Тип тела данных POST. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (2), XML (3).
http_proxy	string	Строка подключения к прокси HTTP(S). Поддерживается для элементов данных HTTP_AGENT.
headers		Корневой элемент для заголовков запроса HTTP(S). Поддерживается для элементов данных HTTP_AGENT.

Элемент	Тип	Описание
name	string	(обязательно для элементов данных HTTP_AGENT) Имя заголовка.
value	string	(обязательно для элементов данных HTTP_AGENT) Значение заголовка.
retrieve_mode	string	Какая часть ответа должна сохраняться. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] BODY (0, по умолчанию), HEADERS (1), BOTH (2).
request_method	string	Тип метода запроса. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] GET (0, по умолчанию), POST (1), PUT (2), HEAD (3).
output_format	string	Способ обработки ответа. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (1).
allow_traps	string	Разрешить заполнение значения так же, как для элемента данных типа trapper. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
ssl_key_password	string	Пароль для файла ключа SSL. Поддерживается для элементов данных HTTP_AGENT.
verify_peer	string	Проверять ли подлинность сертификата узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
verify_host	string	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
time_shift	string	Насколько далеко в прошлое (до текущего времени) заканчивается временной диапазон, который запрашивает элемент данных. Поддерживается для элементов данных TELEMETRY_QUERY.
lookback_limit	string	Насколько далеко в прошлое (до текущего времени) элемент данных запрашивает данные, если отсутствует предыдущий запрос, от которого можно продолжить (например, при первом запуске элемента данных), или если предыдущий запрос старше `delay` (например, из-за недоступности сервера Zabbix). Поддерживается для элементов данных TELEMETRY_QUERY.
granularity	string	Длина фрагмента, на который разбивается полный временной диапазон запроса. Поддерживается для элементов данных TELEMETRY_QUERY.
query		(обязательно для элементов данных TELEMETRY_QUERY) Корневой элемент запроса телеметрии. Определяет, какие данные элемент данных запрашивает из источника данных, как он фильтрует данные и как агрегирует результат.
signal_type	string	(обязательно) Тип сигнала, который запрашивает элемент данных. Поддерживается для элементов данных TELEMETRY_QUERY. Возможные значения:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(обязательно для типа сигнала METRICS) Тип точки метрики, который запрашивает элемент данных. Поддерживается для элементов данных TELEMETRY_QUERY. Возможные значения:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(обязательно) Корневой элемент для столбцов данных телеметрии, которые запрашивает элемент данных. Этот элемент содержит следующие элементы: `column (string)` - (обязательно) Имя столбца данных телеметрии, который запрашивает элемент данных. `attribute_key (string)` - (обязательно для столбцов, содержащих пары «ключ-значение») Имя ключа, который запрашивает элемент данных, в столбце, содержащем пары «ключ-значение».

Элемент	Тип	Описание
aggregated_columns		(обязательно) Корневой элемент для агрегированных столбцов данных телеметрии, которые запрашивает элемент данных. Этот элемент содержит следующие элементы: `column` (`string`) - (обязательно для функций MIN, MAX, AVG, SUM и PERCENTILE) Имя столбца данных телеметрии, который запрашивает элемент данных. `function` (`string`) - (обязательно) Функция, применяемая элементом данных к записям столбца. Возможные значения:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (обязательно для функций PERCENTILE) Процентиль, который элемент данных вычисляет по записям столбца. `alias` (`string`) - (обязательно) Уникальное имя вычисленного значения, используемое для его идентификации в значении элемента данных.
filter		(обязательно) Корневой элемент фильтра запроса телеметрии для записей столбцов в запросе. Этот элемент содержит следующие элементы: `evaltype` (`string`) - (обязательно) Метод вычисления условий фильтра. Возможные значения:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (обязательно для методов вычисления FORMULA) Определяемое пользователем выражение для вычисления условий фильтров с пользовательским выражением. `conditions` (`string`) - (обязательно) Корневой элемент условий фильтра данных телеметрии, которым должна соответствовать запись в столбце данных телеметрии. Элемент `conditions` содержит следующие элементы: `column` (`string`) - (обязательно) Имя столбца данных телеметрии, к которому применяются условия фильтра. `attribute_key` (`string`) - (обязательно для столбцов, содержащих пары «ключ-значение») Имя ключа, к которому применяется условие, в столбце, содержащем пары «ключ-значение». `operator` (`string`) - (обязательно) Оператор условия фильтра. Возможные значения для столбцов с одиночными значениями:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Возможные значения для столбцов, содержащих пары «ключ-значение»:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (обязательно для методов вычисления FORMULA) Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. `value` (`string`) - (обязательно для операций EQUAL, NOT_EQUAL, LIKE и NOT LIKE) Значение для сравнения.
tags		Корневой элемент для тегов элемента данных.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
triggers		Корневой элемент для триггеров элементов данных узла сети.

Note:
См. также: Объект элемента данных (соответствующее свойство с совпадающим именем).

Шаги предварительной обработки значения элемента данных узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
```

```
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

Element	Type	Description
type	string	(обязательно) Тип шага предварительной обработки значения элемента данных. Возможные значения:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, calculated: received value - previous value), CHANGE_PER_SECOND (10, calculated: (received value - previous value)/(time now - time of last check)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(обязательно) Корневой элемент для параметров шага предварительной обработки значения элемента данных.
parameter	string	Отдельный параметр шага предварительной обработки значения элемента данных.
error_handler	string	Тип действия, используемый в случае сбоя шага предварительной обработки. Возможные значения:[1] ORIGINAL_ERROR (0, default), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Параметры обработчика ошибок.

Note:
См. также: Объект предварительной обработки элемента данных (ссылайтесь на соответствующее свойство с совпадающим именем).

Триггеры элемента данных узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL:                  '
              event_name: 'MySQL:                  (           30m)'
              priority: INFO
              description: 'Zabbix                                          30    .'
              dependencies:
                - name: 'MySQL:           '
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL:                '
              event_name: 'MySQL:                (uptime < 10m)'
              priority: INFO
              description: '        MySQL      10    .'
```

```
            tags:
              - tag: scope
                value: notice
      (...)
```

Element	Type	Description
uuid	string	(required) Уникальный идентификатор для этого триггера.
expression	string	(required) Выражение триггера.
recovery_mode	string	Основа для генерации событий OK. Possible values:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Выражение восстановления триггера.
correlation_mode	string	Режим корреляции (без корреляции событий или корреляция событий по тегу). Possible values:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	Имя тега, используемое для корреляции событий.
name	string	(required) Имя триггера.
event_name	string	Имя события.
opdata	string	Оперативные данные.
url_name	string	Метка для URL, связанного с триггером.
url	string	URL, связанный с триггером.
status	string	Статус триггера. Possible values:[1] ENABLED (0, default), DISABLED (1).
priority	string	Серьезность триггера. Possible values:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Описание триггера.
type	string	Тип генерации событий (одно проблемное событие или несколько проблемных событий). Possible values:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	Ручное закрытие проблемных событий. Possible values:[1] NO (0, default), YES (1).
dependencies		Корневой элемент для зависимостей.
name	string	(required) Имя зависимого триггера.
expression	string	(required) Выражение зависимого триггера.
recovery_expression	string	Выражение восстановления зависимого триггера.
tags		Корневой элемент для тегов триггера.
tag	string	(required) Имя тега.
value	string	Значение тега.

Note:
См. также: Trigger object (см. соответствующее свойство с совпадающим именем).

Правила низкоуровневого обнаружения узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: '             '
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: ( .        )
          description: '                  .'
          interface_ref: if1
          item_prototypes:
            - name: '            {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: '           .'
```

```
            preprocessing:
              - type: DISCARD_UNCHANGED_HEARTBEAT
                parameters:
                  - 1h
            interface_ref: if1
            tags:
              - tag: component
                value: storage
              - tag: database
                value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
              path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
Большинство элементов правил низкоуровневого обнаружения узла сети такие же, как и для элементов данных узла сети. В таблице ниже описаны только те элементы, которые отличаются от элементов данных узла сети.

Element	Type	Description
type	string	Тип элемента данных. Возможные значения:[1] ZABBIX_PASSIVE (0, default), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(required) Ключ правила низкоуровневого обнаружения.
filter		Корневой элемент для фильтров правил низкоуровневого обнаружения узла сети.
lifetime	string	Период времени (в секундах, суффикс времени или пользовательский макрос), по истечении которого больше не обнаруженные ресурсы будут удалены. По умолчанию: 7d.
lifetime_type	string	Сценарий удаления потерянных ресурсов LLD. Возможные значения: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Период времени (в секундах, суффикс времени или пользовательский макрос), по истечении которого больше не обнаруженные ресурсы будут отключены.
enabled_lifetime_type	string	Сценарий отключения потерянных ресурсов LLD. Возможные значения: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Корневой элемент для прототипов элементов данных узла сети, которые такие же, как и для элементов данных узла сети.
trigger_prototypes		Корневой элемент для прототипов триггеров узла сети, которые такие же, как и для триггеров элементов данных узла сети.
graph_prototypes		Корневой элемент для прототипов графиков узла сети, которые такие же, как и для графиков узла сети.
host_prototypes		Корневой элемент для прототипов узлов сети, которые такие же, как и для узлов сети.
parent_discovery_rule		Корневой элемент для родительского правила низкоуровневого обнаружения (или прототипа правила) прототипа правила низкоуровневого обнаружения. Это свойство указывает, что это прототип правила LLD, прямой дочерний элемент указанного правила/прототипа правила.
key	string	(required) Ключ родительского правила низкоуровневого обнаружения (или прототипа правила).
master_item	string	(required for DEPENDENT rules) Корневой элемент для master-элемента зависимого правила.
lld_macro_paths		Корневой элемент для путей макросов правила низкоуровневого обнаружения.
lld_macro	string	(required) Имя макроса правила низкоуровневого обнаружения.
path	string	(required) Селектор значения, которое будет присвоено соответствующему макросу.

Element	Type	Description
preprocessing		Корневой элемент для предварительной обработки значения правила низкоуровневого обнаружения.
step		Корневой элемент для элементов шага предварительной обработки значения правила низкоуровневого обнаружения, которые такие же, как и для шагов предварительной обработки значения элемента данных узла сети, за исключением меньшего числа возможных значений. См. также: объект предварительной обработки правила LLD.
overrides		Корневой элемент для правил переопределения правила низкоуровневого обнаружения.
name	string	(required) Уникальное имя переопределения.
step	string	(required) Уникальный порядковый номер переопределения.
stop	string	Остановить обработку следующих переопределений, если совпадает.
filter		Корневой элемент для элементов фильтра правила переопределения правила низкоуровневого обнаружения, которые такие же, как и для фильтров правил низкоуровневого обнаружения узла сети.
operations		Корневой элемент для операций переопределения правил низкоуровневого обнаружения узла сети.

Note:
См. также: объект правила LLD (ссылайтесь на соответствующее свойство с совпадающим именем).

Фильтры правила низкоуровневого обнаружения узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Element	Type	Description
evaltype	string	Переопределяет метод оценки условий фильтра. Возможные значения:[1] AND_OR (0, по умолчанию), AND (1), OR (2), FORMULA (3).
formula	string	Пользовательская формула вычисления для условий фильтра.
conditions		Корневой элемент для условий фильтра.
macro	string	(required) Имя макроса низкоуровневого обнаружения, для которого выполняется проверка.
value	string	Значение, с которым выполняется сравнение.
operator	string	Оператор условия. Возможные значения:[1] MATCHES_REGEX (8, по умолчанию), NOT_MATCHES_REGEX (9).
formulaid	string	(required) Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть задан пользователем при изменении условий фильтра, но будет сгенерирован заново при последующем запросе.

Note:
См. также: объект фильтра правила LLD (см. соответствующее свойство с совпадающим именем).

Операции переопределения правила низкоуровневого обнаружения узла сети

Element	Type	Description
operationobject	string	Объект, к которому применяется операция. Возможные значения:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).
operator	string	Оператор условия переопределения. Возможные значения:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Регулярное выражение или строка для оператора условия переопределения.
status	string	Состояние объекта после операции переопределения.
discover	string	Указывает, добавляется ли объект в результате обнаружения.
delay	string	Интервал обновления, заданный для прототипа элемента данных после операции переопределения.
history	string	Период хранения истории, заданный для прототипа элемента данных после операции переопределения.
trends	string	Период хранения трендов, заданный для прототипа элемента данных после операции переопределения.
severity	string	Уровень серьезности прототипа триггера, заданный после операции переопределения.
tags		Корневой элемент для тегов, заданных для объекта после операции переопределения.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
templates		Корневой элемент для шаблонов, связанных с прототипом узла сети после операции переопределения.
name	string	(обязательно) Имя шаблона.
inventory_mode	string	Режим инвентаризации прототипа узла сети, заданный после операции переопределения.

Note:
См. также: Объект операции переопределения правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Веб-сценарии узла сети

Element	Type	Description
uuid	string	(required) Уникальный идентификатор этого веб-сценария.
name	string	(required) Имя веб-сценария.
delay	string	Частота выполнения веб-сценария (в секундах, суффикс времени или пользовательский макрос). По умолчанию: 1m.
attempts	integer	Количество попыток выполнения шагов веб-сценария. Допустимые значения: 1-10 (по умолчанию: 1).
agent	string	Клиентский агент. Zabbix будет выдавать себя за выбранный браузер. Это полезно, когда веб-сайт возвращает разное содержимое для разных браузеров. По умолчанию: Zabbix.
http_proxy	string	Прокси, который будет использоваться веб-сценарием, в формате: `http://[username[:password]@]proxy.example.com[:port]`
variables		Корневой элемент для переменных веб-сценария, которые могут использоваться в шагах сценария.
name	string	(required) Имя переменной.
value	text	(required) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, которые будут отправляться при выполнении запроса. Заголовки следует перечислять, используя тот же синтаксис, что и в HTTP-протоколе.
name	string	(required) Имя заголовка.
value	text	(required) Значение заголовка.

Element	Type	Description
status	string	Статус веб-сценария. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).
authentication	string	Метод аутентификации. Возможные значения:[1] NONE (0, по умолчанию), BASIC (1), NTLM (2), Kerberos (3) или Digest (4).
http_user	string	Имя пользователя, используемое для аутентификации BASIC (HTTP), NTLM, Kerberos или Digest.
http_password	string	Пароль, используемый для аутентификации BASIC (HTTP), NTLM, Kerberos или Digest.
verify_peer	string	Проверять SSL-сертификат веб-сервера. Возможные значения:[1] NO (0, по умолчанию), YES (1).
verify_host	string	Проверять, что поле *Common Name* или поле *Subject Alternate Name* сертификата веб-сервера совпадает. Возможные значения:[1] NO (0, по умолчанию), YES (1).
ssl_cert_file	string	Имя файла SSL-сертификата, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_file	string	Имя файла закрытого ключа SSL, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль файла закрытого ключа SSL.
steps		(required) Корневой элемент для шагов веб-сценария узла сети.
tags		Корневой элемент для тегов веб-сценария.
tag	string	(required) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект веб-сценария (см. соответствующее свойство с совпадающим именем).

Шаги веб-сценария узла сети

Element	Type	Description
name	string	(required) Имя шага веб-сценария.
url	string	(required) URL для мониторинга.
query_fields		Корневой элемент для параметров запроса (массив HTTP-полей, которые будут добавлены к URL при выполнении запроса).
name	string	(required) Имя параметра запроса.
value	string	Значение параметра запроса.
posts		Корневой элемент для переменных HTTP POST (строка (raw post data) или массив HTTP-полей (данные полей формы)).
name	string	(required) Имя поля POST.
value	string	(required) Значение поля POST.
variables		Корневой элемент переменных уровня шага (макросов), которые должны быть применены после этого шага. Если значение переменной имеет префикс ’regex:’, то его значение извлекается из данных, возвращенных этим шагом, согласно шаблону регулярного выражения, следующему за префиксом ’regex:’
name	string	(required) Имя переменной.
value	text	(required) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, которые будут отправлены при выполнении запроса.
name	string	(required) Имя заголовка.
value	text	(required) Значение заголовка.
follow_redirects	string	Следовать HTTP-перенаправлениям. Возможные значения:[1] NO (0), YES (1, default).
retrieve_mode	string	Режим получения HTTP-ответа. Возможные значения:[1] BODY (0, default), HEADERS (1), BOTH (2).
timeout	string	Тайм-аут (в секундах, time suffix или user macro) выполнения шага. По умолчанию: 15s.
required	string	Текст, который должен присутствовать в ответе (игнорируется, если пусто).

Element	Type	Description
status_codes	string	Список HTTP-кодов состояния, разделенных запятыми, которые принимаются (например, 200-201,210-299; игнорируется, если пусто).

Note:
См. также: Объект шага веб-сценария (см. соответствующее свойство с совпадающим именем).

Графики узла сети

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: ( .         )
    (...)
```

Element	Type	Description
uuid	string	Уникальный идентификатор графика.
name	string	(обязательно) Имя графика.
width	integer	Ширина графика в пикселях. Используется для предварительного просмотра и для круговых/развернутых графиков. Возможные значения:[1] 20-65535 (по умолчанию: 900).
height	integer	Высота графика в пикселях. Используется для предварительного просмотра и для круговых/развернутых графиков. Возможные значения:[1] 20-65535 (по умолчанию: 900).
yaxismin	double	Значение минимума оси Y. Поддерживается для FIXED минимального значения оси Y. По умолчанию: 0.
yaxismax	double	Значение максимума оси Y. Поддерживается для FIXED минимального значения оси X. По умолчанию: 0.
show_work_period	string	Выделять нерабочие часы. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] NO (0), YES (1, по умолчанию).
show_triggers	string	Отображать простые значения триггеров в виде линии. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] NO (0), YES (1, по умолчанию).
type	string	Тип графика. Возможные значения:[1] NORMAL (0, по умолчанию), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Отображать легенду графика. Возможные значения:[1] NO (0), YES (1, по умолчанию).
show_3d	string	Включить 3D-стиль. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] NO (0, по умолчанию), YES (1).
percent_left	double	Показывать линию процентиля для левой оси. Поддерживается для графиков NORMAL. По умолчанию: 0.
percent_right	double	Показывать линию процентиля для правой оси. Поддерживается для графиков NORMAL. По умолчанию: 0.
ymin_type_1	string	Минимальное значение оси Y. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] CALCULATED (0, по умолчанию), FIXED (1), ITEM (2).
ymin_item_1		(обязательно, если `ymin_type_1` установлено в `ITEM`) Корневой элемент для сведений об отдельном элементе данных.
host	string	(обязательно) Узел сети элемента данных.
key	string	(обязательно) Ключ элемента данных.

Element	Type	Description
ymax_type_1	string	Максимальное значение оси Y. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] CALCULATED (0, по умолчанию), FIXED (1), ITEM (2).
ymax_item_1		(обязательно, если `ymax_type_1` установлено в `ITEM`) Корневой элемент для сведений об отдельном элементе данных.
host	string	(обязательно) Узел сети элемента данных.
key	string	(обязательно) Ключ элемента данных.
graph_items		(обязательно) Корневой элемент для элементов графика узла сети.

Note:
См. также: Объект Graph (см. соответствующее свойство с совпадающим именем).

Элементы графика узла сети

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

Элемент	Тип	Описание
sortorder	integer	Порядок отображения. Меньшее значение отображается первым. Может использоваться для отображения линий или областей позади (или перед) другими.
drawtype	string	Стиль отображения элемента графика. Поддерживается для графиков NORMAL. Возможные значения:[1] SINGLE_LINE (0, по умолчанию), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	Цвет элемента (6 символов, hex).
yaxisside	string	Сторона графика, на которой будет отображаться шкала Y элемента графика. Поддерживается для графиков NORMAL и STACKED. Возможные значения:[1] LEFT (0, по умолчанию), RIGHT (1).
calc_fnc	string	Данные для отображения, если для элемента существует более одного значения. Возможные значения:[1] MIN (1), AVG (2, по умолчанию), MAX (4), ALL (7; минимум, среднее и максимум; поддерживается для простых графиков), LAST (9, поддерживается для круговых/развернутых круговых графиков).
type	string	Тип элемента графика. Возможные значения:[1] SIMPLE (0, по умолчанию), GRAPH_SUM (2; значение элемента представляет весь круговой график; поддерживается для круговых/развернутых круговых графиков).
item		(обязательно) Отдельный элемент данных.
host	string	(обязательно) Узел сети элемента данных.
key	string	(обязательно) Ключ элемента данных.

Note:
См. также: Объект элемента графика (см. соответствующее свойство с совпадающим именем).

Карты значений узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(required) Уникальный идентификатор этой карты значений.
name	string	(required) Имя карты значений.
mapping		Корневой элемент для сопоставлений.
type	string	Тип сопоставления значения. Possible values:[1] EQUAL (0, default), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Исходное значение.
newvalue	string	(required) Значение, в которое сопоставляется исходное значение.

Note:
См. также: Объект карты значений (обратитесь к соответствующему свойству с совпадающим именем).

Сноски [1] Целочисленные значения API в скобках, например ENABLED (0), приведены только для справки. Для получения дополнительной информации см. связанную страницу объекта API в записи таблицы или в конце каждого раздела.

Сетевые карты

Обзор

Сетевая карта экспорт содержит:

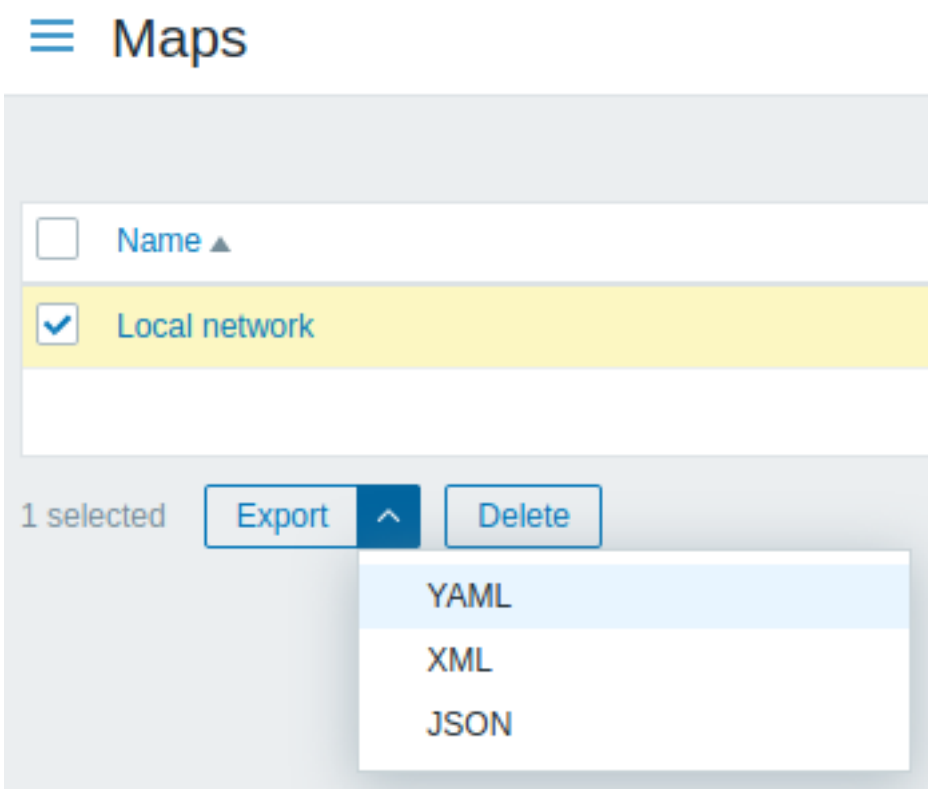
- Все связанные изображения
- Структуру карты (все настройки карты, все содержащиеся элементы с их настройками, связи карты и индикаторы состояния связей карты)

Warning:
Любые группы узлов сети, узлы сети, триггеры, другие карты или другие элементы, которые могут быть связаны с экспортируемой картой, не экспортируются. Таким образом, если отсутствует хотя бы один из элементов, на которые ссылается карта, ее импорт завершится ошибкой.

Экспорт

Чтобы экспортировать карты сети, выполните следующие действия:

1. Перейдите в *Monitoring* > *Maps*.
2. Отметьте флажками карты сети, которые нужно экспортировать.
3. Нажмите *Export* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



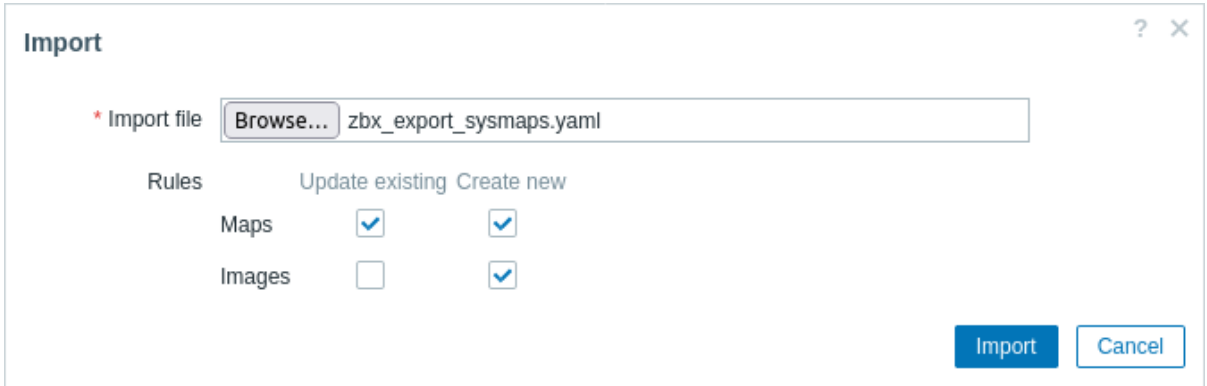
В зависимости от выбранного формата карты экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_maps.yaml` - при экспорте в YAML;
- `zabbix_export_maps.xml` - при экспорте в XML;
- `zabbix_export_maps.json` - при экспорте в JSON.

Импорт

Чтобы импортировать карты сети, выполните следующие действия:

1. Перейдите в раздел *Мониторинг* > *Карты*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл для импорта.
4. Установите флажки правил импорта, которые необходимо применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Правила импорта:

Правило	Описание
Обновлять существующие	Существующие карты будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создавать новые	Новые карты будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.

Если снять флажки обоих параметров карт и установить соответствующие параметры для изображений, будут импортированы только изображения. Импорт изображений доступен только пользователям с ролью *Суперадминистратор*.

Во время импорта нельзя просмотреть изменения карт сети. Сообщение об успешном или неудачном импорте будет отображено в веб-интерфейсе.

Warning:
При замене существующего изображения это повлияет на все карты, использующие это изображение.

Формат экспорта

Экспорт в YAML:

```
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
```

```
            name: Zabbix_server_3D_(128)
          icon_on: {  }
          icon_disabled: {  }
          icon_maintenance: {  }
          urls: {  }
          evaltype: '0'
      shapes:
        - type: '0'
          x: '0'
          'y': '0'
          width: '680'
          height: '15'
          text: '{MAP.NAME}'
          font: '9'
          font_size: '11'
          font_color: '000000'
          text_halign: '0'
          text_valign: '0'
          border_type: '0'
          border_width: '0'
          border_color: '000000'
          background_color: ''
          zindex: '0'
      lines: {  }
      links: {  }
```

Экспортированные элементы

Экспортированные элементы описаны в таблице ниже.

Element	Type	Description
images		Корневой элемент для изображений.
name	string	Уникальное имя изображения.
imagetype	integer	Тип изображения. Возможные значения: 1 - Image; 2 - Background.
encodedImage	string	Изображение, закодированное в Base64.
maps		Корневой элемент для maps.

Карты

Element	Type	Description
name	string	Уникальное имя карты.
width	integer	Ширина карты в пикселях.
height	integer	Высота карты в пикселях.
label_type	integer	Тип метки элемента карты. Возможные значения: 0 - Метка; 1 - IP-адрес узла сети; 2 - Имя элемента; 3 - Только статус; 4 - Ничего.
label_location	integer	Расположение метки элемента карты по умолчанию. Возможные значения: 0 - Снизу; 1 - Слева; 2 - Справа; 3 - Сверху.

Element	Type	Description
highlight	integer	Включить подсветку значков для активных триггеров и статусов узлов сети. Возможные значения: 0 - Нет; 1 - Да.
expandproblem	integer	Показывать проблемный триггер для элементов с одной проблемой. Возможные значения: 0 - Нет; 1 - Да.
markelements	integer	Подсвечивать элементы карты, у которых недавно изменился статус. Возможные значения: 0 - Нет; 1 - Да.
show_unack	integer	Отображение проблем. Возможные значения: 0 - Количество всех проблем; 1 - Количество непризнанных проблем; 2 - Количество признанных и непризнанных проблем отдельно.
severity_min	integer	Минимальная серьезность триггера, отображаемого на карте по умолчанию. Возможные значения: 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа.
show_suppressed	integer	Показывать проблемы, которые в противном случае были бы подавлены (не отображались) из-за обслуживания узла сети. Возможные значения: 0 - Нет; 1 - Да.
grid_size	integer	Размер ячейки сетки карты в пикселях. Поддерживается, если `grid_show` установлено в 0. Возможные значения: 20, 40, 50, 75 или 100.
grid_show	integer	Отображать сетку в настройках карты. Возможные значения: 0 - Да; 1 - Нет.
grid_align	integer	Автоматически выравнивать значки в настройках карты. Возможные значения: 0 - Да; 1 - Нет.
label_format	integer	Использовать расширенную настройку меток. Возможные значения: 0 - Нет; 1 - Да.
label_type_host	integer	Отображать метку как метку узла сети. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Метка; 1 - IP-адрес узла сети; 2 - Имя элемента; 3 - Только статус; 4 - Ничего; 5 - Пользовательская метка.
label_type_hostgroup	integer	Отображать метку как метку группы узлов сети. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Метка; 2 - Имя элемента; 3 - Только статус; 4 - Ничего; 5 - Пользовательская метка.

Element	Type	Description
label_type_trigger	integer	Отображать метку как метку триггера. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Метка; 2 - Имя элемента; 3 - Только статус; 4 - Ничего; 5 - Пользовательская метка.
label_type_map	integer	Отображать метку как метку карты. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Метка; 2 - Имя элемента; 3 - Только статус; 4 - Ничего; 5 - Пользовательская метка.
label_type_image	integer	Отображать метку как метку изображения. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Метка; 2 - Имя элемента; 4 - Ничего; 5 - Пользовательская метка.
label_string_host	string	Пользовательская метка для элементов узла сети. Поддерживается, если `label_type_host` установлено в 5.
label_string_hostgroup	string	Пользовательская метка для элементов группы узлов сети. Поддерживается, если `label_type_hostgroup` установлено в 5.
label_string_trigger	string	Пользовательская метка для элементов триггера. Поддерживается, если `label_type_trigger` установлено в 5.
label_string_map	string	Пользовательская метка для элементов карты. Поддерживается, если `label_type_map` установлено в 5.
label_string_image	string	Пользовательская метка для элементов изображения. Поддерживается, если `label_type_image` установлено в 5.
expand_macros	integer	Разворачивать макросы в метках в настройках карты. Возможные значения: 0 - Нет; 1 - Да.
background		Корневой элемент для фонового изображения (если есть). Поддерживается, если `imagetype` установлено в 2.
name	string	Имя фонового изображения.
iconmap		Корневой элемент для сопоставления значков (если есть).
name	string	Имя сопоставления значков.
urls		Корневой элемент для URL, используемых картами или каждым элементом карты.
name	string	Имя ссылки.
url	string	URL ссылки.
elementtype	integer	Тип элемента карты, к которому относится ссылка. Возможные значения: 0 - Узел сети; 1 - Карта; 2 - Триггер; 3 - Группа узлов сети; 4 - Изображение.
selements		Корневой элемент для элементов карты.
shapes		Корневой элемент для фигур карты.
type	integer	Тип фигуры. Возможные значения: 0 - Прямоугольник; 1 - Эллипс.
x	integer	Координата X фигуры в пикселях.
y	integer	Координата Y фигуры в пикселях.
width	integer	Ширина фигуры.

Element		Type	Description
	height	integer	Высота фигуры.
	text	string	Текст внутри фигуры.
	font	integer	Стиль шрифта текста. Возможные значения: 0 - Georgia, serif; 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif; 2 - ”Times New Roman”, Times, serif; 3 - Arial, Helvetica, sans-serif; 4 - ”Arial Black”, Gadget, sans-serif; 5 - ”Comic Sans MS”, cursive, sans-serif; 6 - Impact, Charcoal, sans-serif; 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif; 8 - Tahoma, Geneva, sans-serif; 9 - ”Trebuchet MS”, Helvetica, sans-serif; 10 - Verdana, Geneva, sans-serif; 11 - ”Courier New”, Courier, monospace; 12 - ”Lucida Console”, Monaco, monospace.
	font_size	integer	Размер шрифта в пикселях.
	font_color	string	Цвет шрифта, представленный в шестнадцатеричном коде.
	text_halign	integer	Горизонтальное выравнивание текста. Возможные значения: 0 - По центру; 1 - Слева; 2 - Справа.
	text_valign	integer	Вертикальное выравнивание текста. Возможные значения: 0 - По середине; 1 - Сверху; 2 - Снизу.
	border_type	integer	Тип границы фигуры. Возможные значения: 0 - Нет; 1 - Жирная линия; 2 - Пунктирная линия; 3 - Штриховая линия.
	border_width	integer	Ширина границы в пикселях.
	border_color	string	Цвет границы, представленный в шестнадцатеричном коде.
	background_color	string	Цвет фона (заливки), представленный в шестнадцатеричном коде.
	zindex	integer	Значение для упорядочивания всех фигур и линий (z-index).
lines			Корневой элемент для линий карты.
	x1	integer	Координата X точки 1 линии в пикселях.
	y1	integer	Координата Y точки 1 линии в пикселях.
	x2	integer	Координата X точки 2 линии в пикселях.
	y2	integer	Координата Y точки 2 линии в пикселях.
	line_type	integer	Тип линии. Возможные значения: 0 - Нет; 1 - Жирная линия; 2 - Пунктирная линия; 3 - Штриховая линия.
	line_width	integer	Ширина линии в пикселях.
	line_color	string	Цвет линии, представленный в шестнадцатеричном коде.
	zindex	integer	Значение для упорядочивания всех фигур и линий (z-index).
links			Корневой элемент для ссылок между элементами карты.
	drawtype	integer	Стиль ссылки. Возможные значения: 0 - Линия; 2 - Жирная линия; 3 - Пунктирная линия; 4 - Штриховая линия.
	color	string	Цвет ссылки (6 символов, hex).

Element	Type	Description
label	string	Метка ссылки.
selementid1	id	ID одного элемента для соединения.
selementid2	id	ID другого элемента для соединения.
linktriggers		Корневой элемент для индикаторов состояния ссылок.

Note:
См. также: Объект карты (см. соответствующее свойство с совпадающим именем).

Элементы карты

Element	Type	Description
elementtype	integer	Тип элемента карты. Возможные значения: 0 - Узел сети; 1 - Карта; 2 - Триггер; 3 - Группа узлов сети; 4 - Изображение.
elements		Корневой элемент для сущностей Zabbix (узел сети, группа узлов сети, карта и т. д.), которые отображаются на карте. У каждой сущности есть соответствующий элемент (`host` и т. д.; см., например, Export format).
label	string	Подпись значка.
label_location	integer	Расположение подписи. Возможные значения: -1 - Использовать значение по умолчанию для карты; 0 - Снизу; 1 - Слева; 2 - Справа; 3 - Сверху.
x	integer	Положение по оси X.
y	integer	Положение по оси Y.
elementsubtype	integer	Подтип элемента. Поддерживается, если для `elementtype` задано значение 3. Возможные значения: 0 - Одна группа узлов сети; 1 - Все группы узлов сети.
areatype	integer	Размер области. Поддерживается, если для `elementtype` задано значение 1. Возможные значения: 0 - Такой же, как у всей карты; 1 - Пользовательский размер.
width	integer	Ширина области. Поддерживается, если для `areatype` задано значение 1.
height	integer	Высота области. Поддерживается, если для `areatype` задано значение 1.
viewtype	integer	Алгоритм размещения области. Поддерживается, если для `elementsubtype` задано значение 1. Возможные значения: 0 - Равномерно размещать в области.
use_iconmap	integer	Использовать сопоставление значков для этого элемента. Актуально только если сопоставление значков включено на уровне карты. Возможные значения: 0 - Нет; 1 - Да.
selementid	id	Уникальный ID записи элемента.
icon_off		Корневой элемент для изображения, используемого, когда элемент находится в состоянии `OK`.
name	string	Уникальное имя изображения.

Element	Type	Description
icon_on		Корневой элемент для изображения, используемого, когда элемент находится в состоянии `Problem`.
name	string	Уникальное имя изображения.
icon_disabled		Корневой элемент для изображения, используемого, когда элемент отключен.
name	string	Уникальное имя изображения.
icon_maintenance		Корневой элемент для изображения, используемого, когда элемент находится на обслуживании.
name	string	Уникальное имя изображения.
urls		Корневой элемент для URL, используемых картами или каждым элементом карты.
name	string	Имя ссылки.
url	string	URL ссылки.
evaltype	integer	Тип оценки для тегов.
tags		Корневой элемент для тегов проблемы (для элементов узла сети и группы узлов сети). Если указаны теги, на карте будут отображаться только проблемы с этими тегами.
tag	string	Имя тега.
value	string	Значение тега.
operator	integer	Оператор.

Note:
См. также: Map element object (обратитесь к соответствующему свойству с совпадающим именем).

Индикаторы состояния связей на карте

Элемент	Тип	Описание
drawtype	integer	Стиль линии связи, когда триггер находится в состоянии ’Problem’. Возможные значения: 0 - Линия; 2 - Жирная линия; 3 - Пунктирная линия; 4 - Штриховая линия.
color	string	Цвет линии связи (6 символов, hex), когда триггер находится в состоянии ’Problem’.
trigger		Корневой элемент для триггера, используемого для отображения состояния связи.
description	string	Имя триггера.
expression	string	Выражение триггера.
recovery_expression	string	Выражение восстановления триггера.

Note:
См. также: Объект триггера связи карты (см. соответствующее свойство с совпадающим именем).

Типы медиа

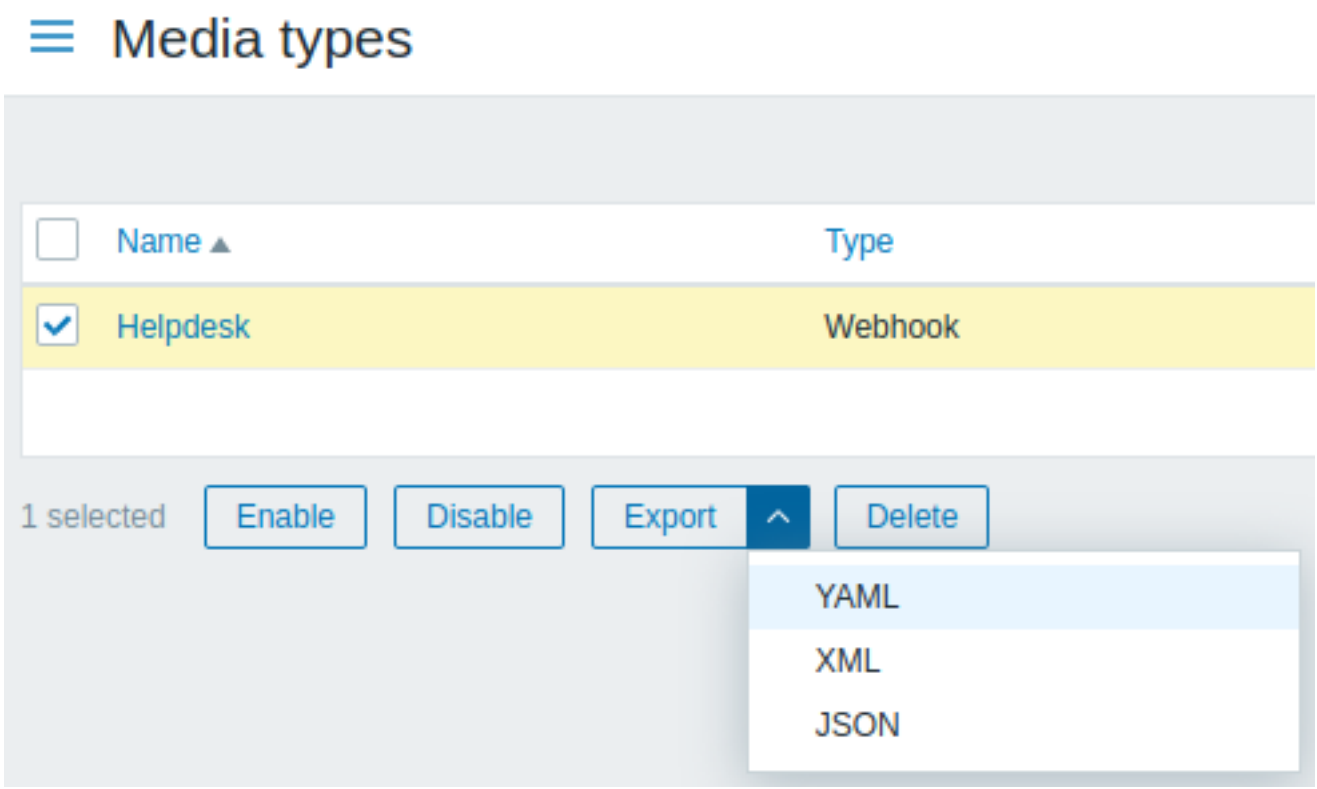
Обзор

Типы медиа экспортируются со всеми связанными объектами и связями между объектами.

Экспорт

Чтобы экспортировать типы оповещений, выполните следующие действия:

1. Перейдите в *Оповещения* > *Типы оповещений*.
2. Отметьте флажками типы оповещений, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



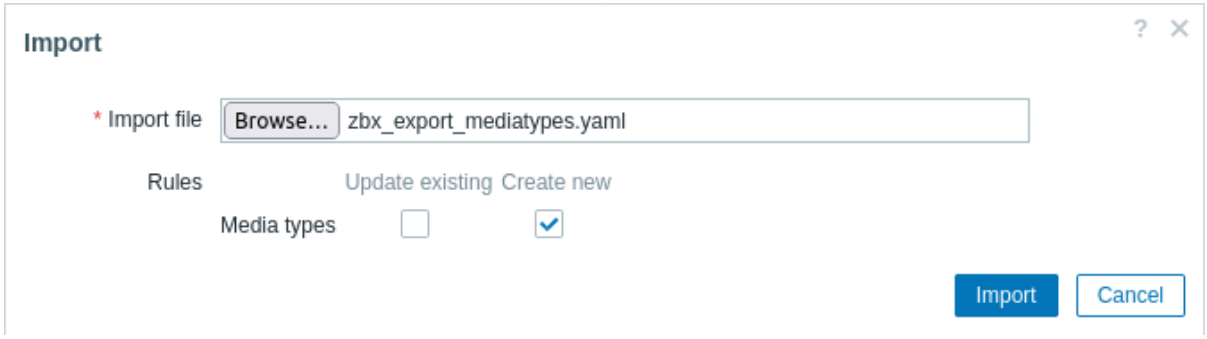
В зависимости от выбранного формата типы оповещений экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_mediatypes.yaml` - при экспорте в YAML;
- `zabbix_export_mediatypes.xml` - при экспорте в XML;
- `zabbix_export_mediatypes.json` - при экспорте в JSON.

Импорт

Чтобы импортировать типы носителей, выполните следующие действия:

1. Перейдите в раздел *Оповещения* > *Типы носителей*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл импорта.
4. Установите флажки для правил импорта, которые необходимо применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Правила импорта:

Правило	Описание
Обновлять существующие	Установите этот флажок, чтобы обновить существующие типы носителей с использованием данных из файла импорта.
Создавать новые	Установите этот флажок, чтобы создать новые типы носителей с использованием данных из файла импорта. Если файл импорта содержит уже существующий тип носителя (например, *Email*), новый тип создан не будет, а существующий тип носителя также не будет обновлен.

Во время импорта нельзя просмотреть изменения типов носителей. Сообщение об успешном или неудачном импорте будет отображено в веб-интерфейсе.

Формат экспорта

Экспорт в YAML:

```
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
```

```
            {name: 'average', color: '#FFA059'},
            {name: 'high', color: '#E97659'},
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw '               "event_source": "' + params.event_source + '".\n           : 0-3.';
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw '               "event_value": ' + params.event_value + '\n           : 0    1.';
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority"               -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw '   "triggerid"            ';
    }

    if (isNaN(params.eventid)) {
        throw '   "eventid"           ';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw '   "message"             ';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw '   "retry"                          30,       "priority"          2';
```

```
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw '   "expire"                          10800,       "priority"          2';
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    .                  : https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/media/pus

                token            Pushover.
                 Pushover        Zabbix                      "        ".
message_templates:
  - event_source: TRIGGERS
    operation_mode: PROBLEM
    subject: '     : {EVENT.NAME}'
    message: |
                    {EVENT.TIME} {EVENT.DATE}
               : {EVENT.NAME}
              : {HOST.NAME}
               : {EVENT.SEVERITY}
                    : {EVENT.OPDATA}
             ID      : {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
```

```
      operation_mode: RECOVERY
      subject: '       {EVENT.DURATION}: {EVENT.NAME}'
      message: |
                       {EVENT.RECOVERY.TIME} {EVENT.RECOVERY.DATE}
                : {EVENT.NAME}
                     : {EVENT.DURATION}
               : {HOST.NAME}
                : {EVENT.SEVERITY}
               ID      : {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: '               {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} {EVENT.UPDATE.ACTION}        {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
        {EVENT.UPDATE.MESSAGE}

                         : {EVENT.STATUS},     : {EVENT.AGE},        : {EVENT.ACK.STATUS}.
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: '       : {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
                    : {DISCOVERY.RULE.NAME}

        IP       : {DISCOVERY.DEVICE.IPADDRESS}
        DNS       : {DISCOVERY.DEVICE.DNS}
                    : {DISCOVERY.DEVICE.STATUS}
                      : {DISCOVERY.DEVICE.UPTIME}

                     : {DISCOVERY.SERVICE.NAME}
                     : {DISCOVERY.SERVICE.PORT}
                        : {DISCOVERY.SERVICE.STATUS}
                          : {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: '               : {HOST.HOST}'
      message: |
                : {HOST.HOST}
        IP       : {HOST.IP}
               : {HOST.PORT}
```

Экспортируемые элементы

Экспортируемые элементы описаны в таблице ниже.

Element	Type	Description
name	string	(required) Имя типа оповещений.
type	string	(required) Транспорт, используемый типом оповещений. Possible values:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Включен ли тип оповещений. Possible values:[1] ENABLED (0, default), DISABLED (1)
max_sessions	integer	Максимальное число оповещений, которые могут обрабатываться параллельно. Possible values for SMS:[1] 1 (default). Possible values for other media types:[1] 0-100 (where 0 - unlimited).
attempts	integer	Максимальное число попыток отправки оповещения. Possible values:[1] 1-10 (default: 3).
attempt_interval	string	Интервал (в секундах или с использованием суффикса времени) между повторными попытками. Possible values:[1] 0-60s (default: 10s).
description	string	Описание типа оповещений.
message_templates		Корневой элемент для шаблонов сообщений типа оповещений.

Element	Type	Description
event_source	string	(required) Источник события. Possible values:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Режим работы. Possible values:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Тема сообщения.
message	string	Текст сообщения.

Note:
См. также: Объект Media type (обратитесь к соответствующему свойству с совпадающим именем).

Email

Следующие дополнительные элементы экспортируются только для типа медиа *Email*.

Element	Type	Description
provider	string	Поставщик электронной почты.
smtp_server	string	SMTP-сервер.
smtp_port	integer	Порт SMTP-сервера для подключения. По умолчанию: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	Адрес электронной почты, с которого будут отправляться уведомления.
smtp_security	string	Уровень безопасности SMTP-соединения для использования. Возможные значения:[1] NONE (0, по умолчанию), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Проверка SSL host для SMTP. Возможные значения:[1] NO (0, по умолчанию), YES (1).
smtp_verify_peer	string	Проверка SSL peer для SMTP. Возможные значения:[1] NO (0, по умолчанию), YES (1).
smtp_authentication	string	Метод SMTP-аутентификации для использования. Возможные значения:[1] NONE (0, по умолчанию), PASSWORD (1), OAUTH (2).
username	string	Имя пользователя.
password	string	Пароль аутентификации.
redirection_url	string	URL веб-интерфейса Zabbix для возврата после авторизации OAuth.
client_id	string	Идентификатор клиента, зарегистрированный на сервере авторизации OAuth.
authorization_url	string	URL OAuth с параметрами для получения токенов доступа и обновления.
token_url	string	URL OAuth для обмена токена авторизации на токены доступа и обновления.
message_format	string	Формат сообщения. Возможные значения:[1] TEXT (0), HTML (1, по умолчанию).

Note:
См. также: Объект media type (обратитесь к соответствующему свойству с совпадающим именем).

SMS

Следующие дополнительные элементы экспортируются только для типа медиа *SMS*.

Element	Type	Description
gsm_modem	string	(required) Имя последовательного устройства GSM-модема.

Note:
См. также: Объект media type (см. соответствующее свойство с совпадающим именем).

Скрипт

Следующие дополнительные элементы экспортируются только для типа медиа *Script*.

Element	Type	Description
script	string	(required) Имя скрипта.
name		
parameters		Корневой элемент для параметров скрипта.
sortorder	string	(required) Порядок параметров скрипта, передаваемых в скрипт как аргументы командной строки.
value	string	Значение параметра скрипта.

Note:
См. также: Объект media type (см. соответствующее свойство с совпадающим именем).

Вебхук

Следующие дополнительные элементы экспортируются только для типа медиа *Вебхук*.

Element	Type	Description
script	string	Скрипт.
timeout	string	Интервал тайм-аута HTTP-запроса скрипта Javascript. Possible values:[1] 1-60s (default: 30s).
process_tags	string	Следует ли обрабатывать возвращаемые теги. Possible values:[1] NO (0, default), YES (1).
show_event_menu	string	Указывает на наличие пункта в меню события, если макрос {EVENT.TAGS.*} был успешно разрешен в полях `event_menu_url` и `event_menu_name`. Possible values:[1] NO (0, default), YES (1).
event_menu_url	string	URL пункта меню события. Поддерживает макрос {EVENT.TAGS.*}.
event_menu_name	string	Имя пункта меню события. Поддерживает макрос {EVENT.TAGS.*}.
parameters		Корневой элемент для параметров типа медиа webhook.
name	string	(required) Имя параметра webhook.
value	string	Значение параметра webhook.

Note:
См. также: Объект типа медиа (см. соответствующее свойство с совпадающим именем).

Сноски

[1] Целочисленные значения API в скобках, например ENABLED (0), приведены только для справки. Для получения дополнительной информации см. связанную страницу объекта API в строке таблицы или в конце каждого раздела.

Панели мониторинга

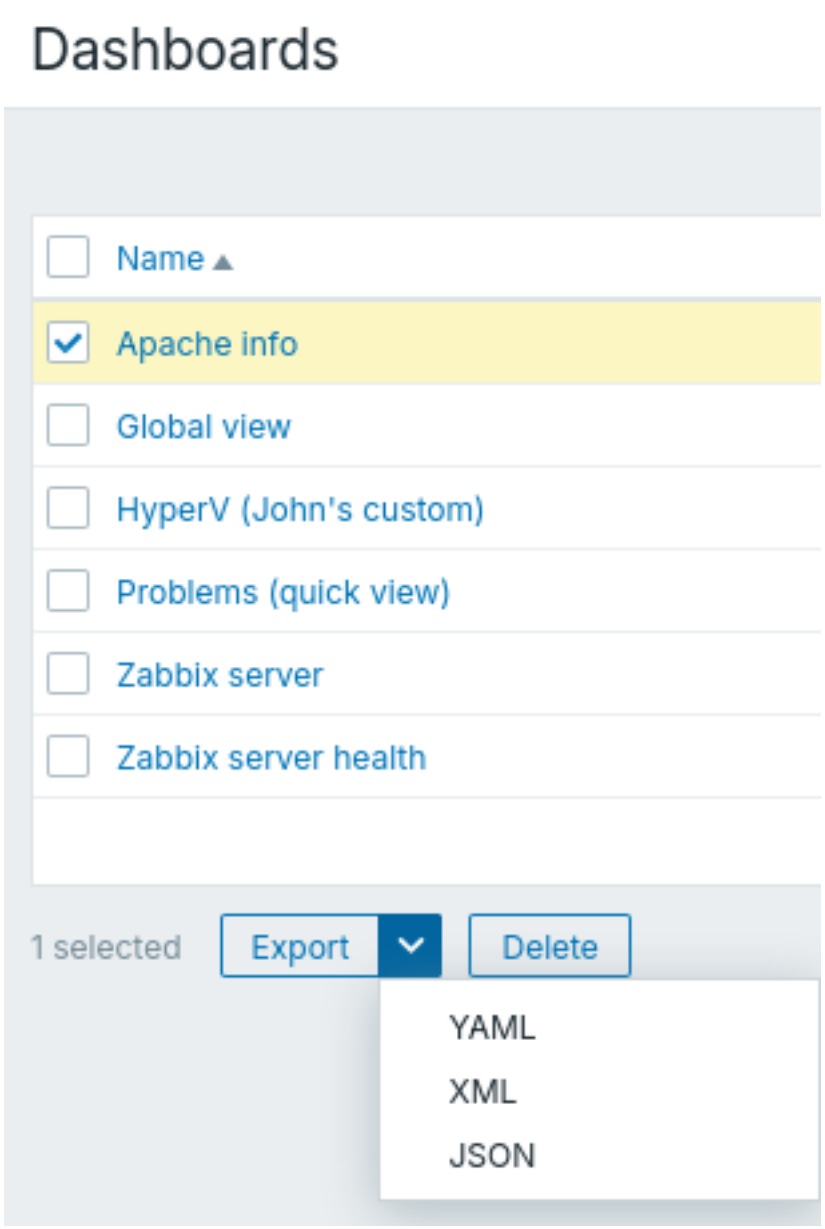
Обзор

Панели мониторинга экспортируются со всеми связанными объектами и связями между объектами.

Экспорт

Чтобы экспортировать панели, выполните следующие действия:

1. Перейдите в *Панели* > *Все панели*.
2. Отметьте флажками панели, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



В зависимости от выбранного формата панели экспортируются в локальный файл с именем по умолчанию:

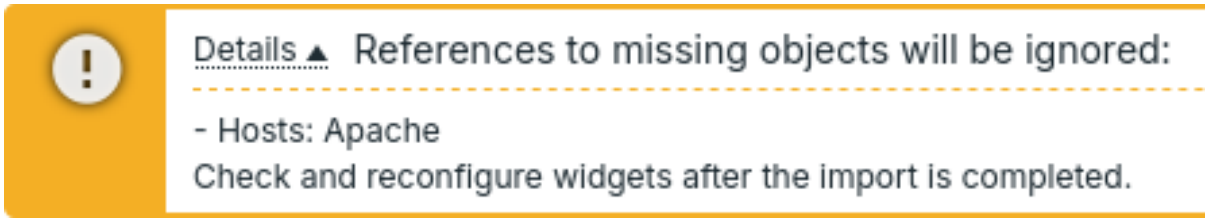
- `zbx_export_dashboards.yaml` - при экспорте в YAML;
- `zbx_export_dashboards.xml` - при экспорте в XML;
- `zbx_export_dashboards.json` - при экспорте в JSON.

Импорт

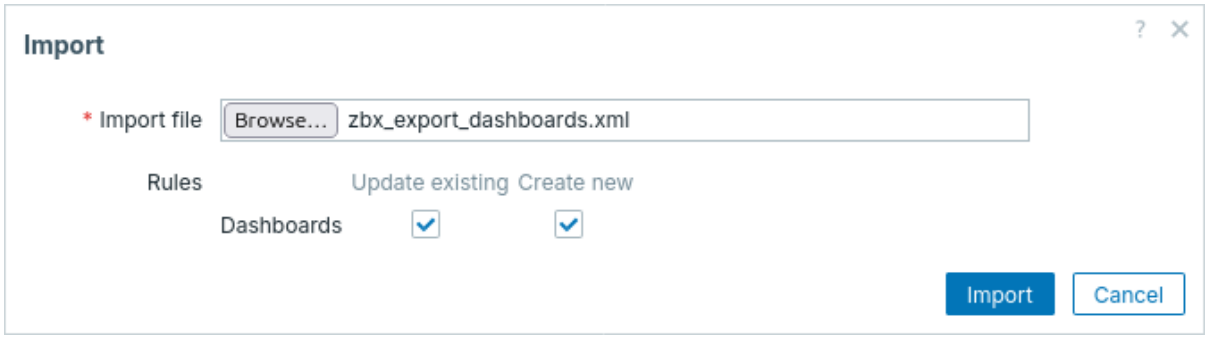
Чтобы импортировать панели мониторинга, выполните следующие действия:

1. Перейдите в раздел *Панели мониторинга* > *Все панели мониторинга*.
2. Нажмите *Импорт* в правом верхнем углу. Кнопка отключена, если у пользователя нет разрешения *Создание и редактирование панелей мониторинга*.
3. Выберите файл импорта.
4. Установите флажки правил импорта, которые необходимо применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.

Если файл импорта содержит ссылки на объекты, отсутствующие в целевой системе, появится предупреждение. Это ожидаемо для виджетов, ссылающихся на определенные узлы сети, элементы данных или другие объекты. В таких случаях Zabbix позволяет импортировать панель мониторинга, удаляет отсутствующие ссылки и либо помечает виджет как настроенный не полностью, если данные обязательны, либо удаляет критерии фильтрации, если поля используются только для фильтрации.



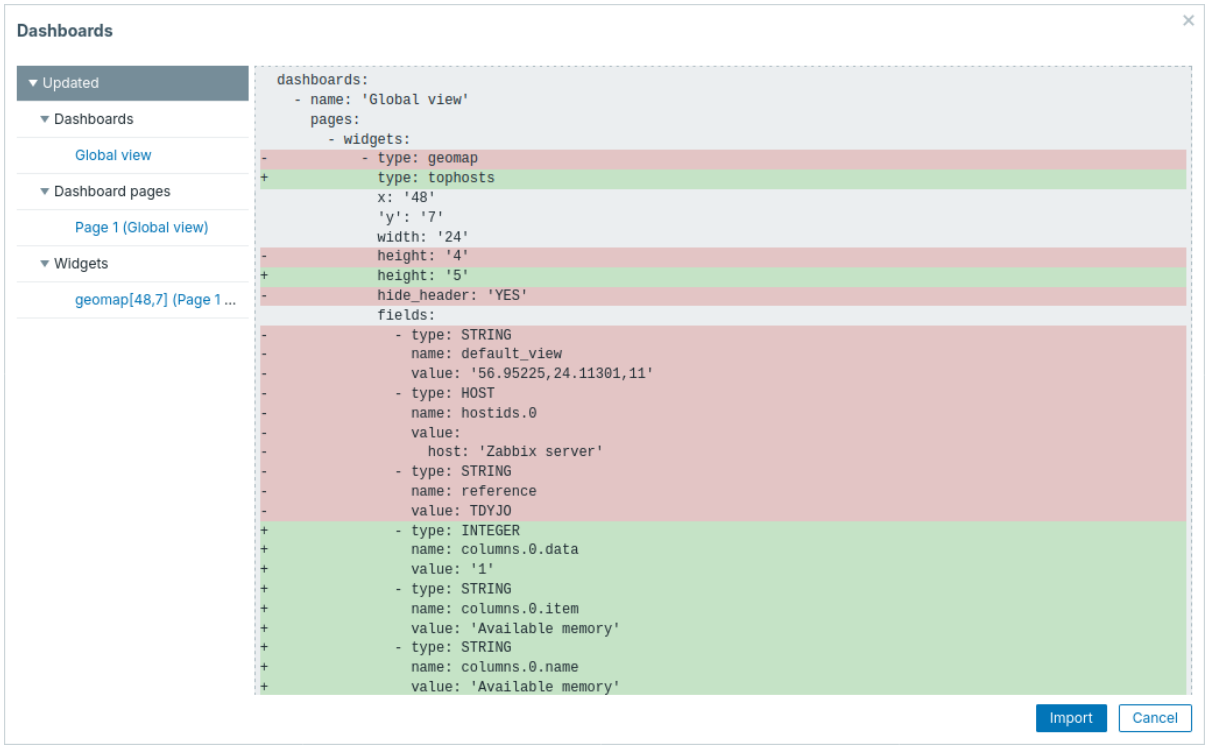
Повторное нажатие кнопки *Импорт* добавит и/или обновит панель мониторинга.



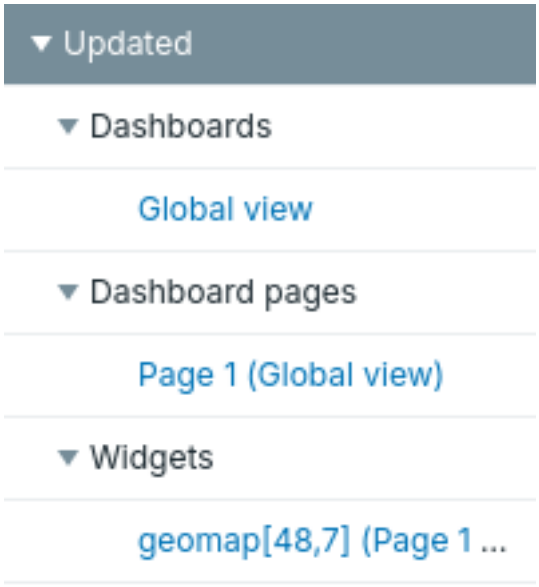
Правила импорта:

Правило	Описание
Обновлять существующие	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создавать новые	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.

На следующем экране можно просмотреть содержимое импортируемых панелей мониторинга. Если это новая панель мониторинга, все элементы будут перечислены на зеленом фоне. При обновлении существующей панели мониторинга новые элементы панели мониторинга будут выделены зеленым цветом, удаленные элементы панели мониторинга — красным, а неизменившиеся элементы будут перечислены на сером фоне.



Меню слева можно использовать для навигации по списку изменений. В разделе *Обновлено* выделены все изменения, внесенные в существующие элементы панели мониторинга. В разделе *Добавлено* перечислены новые элементы панели мониторинга. Элементы в каждом разделе сгруппированы по типу элемента; нажмите серую стрелку, чтобы развернуть или свернуть группу элементов.



Просмотрите изменения панели мониторинга, затем нажмите *Импорт*, чтобы выполнить импорт панели мониторинга. В веб-интерфейсе появится сообщение об успешном выполнении или ошибке импорта.

Формат экспорта

Экспорт в YAML:

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: '                CPU'
              width: '25'
              height: '4'
              fields:
                - type: INTEGER
                  name: column
                  value: '1'
                - type: INTEGER
                  name: columns.0.data
                  value: '2'
                - type: STRING
                  name: columns.0.item
                  value: 'System name'
                - type: STRING
                  name: columns.0.name
                  value: 'Host name'
                - type: STRING
                  name: columns.1.base_color
                  value: 4CAF50
                - type: INTEGER
                  name: columns.1.data
                  value: '1'
                - type: INTEGER
                  name: columns.1.display
                  value: '3'
                - type: STRING
                  name: columns.1.item
                  value: 'CPU utilization'
```

```
      - type: STRING
        name: columns.1.max
        value: '100'
      - type: STRING
        name: columns.1.min
        value: '0'
      - type: STRING
        name: columns.1.name
        value: Utilization
      - type: INTEGER
        name: columns.2.data
        value: '1'
      - type: STRING
        name: columns.2.item
        value: 'Load average (1m avg)'
      - type: STRING
        name: columns.2.name
        value: '1m avg'
      - type: INTEGER
        name: columns.3.data
        value: '1'
      - type: STRING
        name: columns.3.item
        value: 'Load average (5m avg)'
      - type: STRING
        name: columns.3.name
        value: '5m avg'
      - type: INTEGER
        name: columns.4.data
        value: '1'
      - type: STRING
        name: columns.4.item
        value: 'Load average (15m avg)'
      - type: STRING
        name: columns.4.name
        value: '15m avg'
      - type: INTEGER
        name: columns.5.data
        value: '1'
      - type: INTEGER
        name: columns.5.decimal_places
        value: '0'
      - type: STRING
        name: columns.5.item
        value: 'Number of processes'
      - type: STRING
        name: columns.5.name
        value: Processes
      - type: STRING
        name: columnsthresholds.1.color.0
        value: FFFF00
      - type: STRING
        name: columnsthresholds.1.color.1
        value: FF8000
      - type: STRING
        name: columnsthresholds.1.color.2
        value: FF4000
      - type: STRING
        name: columnsthresholds.1.threshold.0
        value: '50'
      - type: STRING
        name: columnsthresholds.1.threshold.1
```

```
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
        - type: hostavail
          'y': '4'
          width: '25'
          fields:
            - type: INTEGER
              name: interface_type.0
              value: '1'
        - type: problems
          name: '          '
          'y': '6'
          width: '48'
          height: '5'
          fields:
            - type: STRING
              name: reference
              value: BRUQE
            - type: INTEGER
              name: show
              value: '3'
            - type: INTEGER
              name: show_tags
              value: '3'
        - type: systeminfo
          x: '25'
          width: '23'
          height: '4'
        - type: problemsbysv
          name: '                  '
          x: '25'
          'y': '4'
          width: '23'
          fields:
            - type: STRING
              name: reference
              value: LBSVB
            - type: INTEGER
              name: show_type
              value: '1'
        - type: gauge
          name: '    Zabbix'
          x: '48'
          width: '12'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: angle
              value: '270'
            - type: INTEGER
              name: desc_size
              value: '8'
            - type: ITEM
              name: itemid.0
              value:
                host: 'Zabbix server'
```

```
    key: vm.memory.utilization
- type: STRING
  name: max
  value: '100'
- type: STRING
  name: min
  value: '0'
- type: INTEGER
  name: scale_size
  value: '11'
- type: INTEGER
  name: show.0
  value: '1'
- type: INTEGER
  name: show.1
  value: '2'
- type: INTEGER
  name: show.2
  value: '3'
- type: INTEGER
  name: show.3
  value: '4'
- type: INTEGER
  name: show.4
  value: '5'
- type: STRING
  name: thresholds.0.color
  value: 42A5F5
- type: STRING
  name: thresholds.0.threshold
  value: '0'
- type: STRING
  name: thresholds.1.color
  value: 9CCC65
- type: STRING
  name: thresholds.1.threshold
  value: '20'
- type: STRING
  name: thresholds.2.color
  value: FFEE58
- type: STRING
  name: thresholds.2.threshold
  value: '40'
- type: STRING
  name: thresholds.3.color
  value: FFA726
- type: STRING
  name: thresholds.3.threshold
  value: '60'
- type: STRING
  name: thresholds.4.color
  value: FF7043
- type: STRING
  name: thresholds.4.threshold
  value: '80'
- type: INTEGER
  name: th_arc_size
  value: '9'
- type: INTEGER
  name: th_show_arc
  value: '1'
- type: INTEGER
```

```
            name: th_show_labels
            value: '1'
          - type: INTEGER
            name: units_bold
            value: '1'
          - type: INTEGER
            name: units_size
            value: '24'
          - type: INTEGER
            name: value_arc_size
            value: '30'
          - type: INTEGER
            name: value_bold
            value: '1'
          - type: INTEGER
            name: value_size
            value: '24'
      - type: svggraph
        x: '48'
        'y': '4'
        width: '24'
        height: '3'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: axisx
            value: '0'
          - type: STRING
            name: ds.0.color
            value: AB47BC
          - type: INTEGER
            name: ds.0.fill
            value: '0'
          - type: STRING
            name: ds.0.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
```

```
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
        - type: geomap
          x: '48'
          'y': '7'
          width: '24'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: STRING
              name: default_view
              value: '56.95225,24.11301,11'
            - type: HOST
              name: hostids.0
              value:
                host: 'Zabbix server'
            - type: STRING
              name: reference
              value: TDYJO
        - type: clock
          name: '        '
          x: '60'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: clock_type
              value: '1'
            - type: INTEGER
              name: show.0
              value: '1'
            - type: INTEGER
              name: show.1
              value: '2'
            - type: INTEGER
              name: show.2
              value: '3'
            - type: INTEGER
              name: time_sec
              value: '0'
            - type: INTEGER
              name: tzone_format
              value: '1'
        - type: item
          name: '               '
          x: '60'
          'y': '2'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
```

```
      name: aggregate_function
      value: '2'
    - type: INTEGER
      name: decimal_size
      value: '32'
    - type: STRING
      name: description
      value: 'Values per second'
    - type: INTEGER
      name: desc_h_pos
      value: '0'
    - type: INTEGER
      name: desc_v_pos
      value: '0'
    - type: ITEM
      name: itemid.0
      value:
        host: 'Zabbix server'
        key: 'zabbix[wcache,values]'
    - type: INTEGER
      name: show.0
      value: '1'
    - type: INTEGER
      name: show.1
      value: '2'
    - type: INTEGER
      name: show.2
      value: '5'
    - type: INTEGER
      name: sparkline.fill
      value: '2'
    - type: INTEGER
      name: sparkline.width
      value: '2'
    - type: STRING
      name: thresholds.0.color
      value: DCEDC8
    - type: STRING
      name: thresholds.0.threshold
      value: '20'
    - type: STRING
      name: thresholds.1.color
      value: FFF9C4
    - type: STRING
      name: thresholds.1.threshold
      value: '40'
    - type: STRING
      name: thresholds.2.color
      value: FFCCBC
    - type: STRING
      name: thresholds.2.threshold
      value: '60'
    - type: INTEGER
      name: units_bold
      value: '0'
    - type: INTEGER
      name: units_show
      value: '0'
    - type: INTEGER
      name: value_h_pos
      value: '0'
    - type: INTEGER
```

```
                name: value_size
                value: '32'
```

Экспортированные элементы

Экспортированные элементы описаны в таблице ниже.

Element		Type	Description
name		string	(required) Название панели мониторинга.
pages			Корневой элемент для страниц панели мониторинга
	widgets		Корневой элемент для страниц панели мониторинга

Note:
См. также: Объект Dashboard (см. соответствующее свойство с совпадающим именем).

Виджет панели

Element	Type	Description
type	string	(required) Тип виджета панели. Возможные значения: actionlog - Журнал действий; clock - Часы; discovery - Статус обнаружения; favgraphs - Избранные графики; favmaps - Избранные карты; gauge - Индикатор; geomap - Геокарта; graph - График (классический); graphprototype - Прототип графика; honeycomb - Соты; hostavail - Доступность узла сети; hostcard - Карточка узла сети; hostnavigator - Навигатор узлов сети; itemcard - Карточка элемента данных; itemhistory - История элемента данных; itemnavigator - Навигатор элементов данных; item - Значение элемента данных; map - Карта; navtree - Дерево навигации по карте; piechart - Круговая диаграмма; problemhosts - Узлы сети с проблемами; problems - Проблемы; problemsbysv - Проблемы по уровню серьезности; scatterplot - Точечная диаграмма; slareport - Отчет SLA; svggraph - График; systeminfo - Системная информация; tophosts - Топ узлов сети; topitems - Топ элементов данных; toptriggers - Топ триггеров; trigover - Обзор триггеров; url - URL; web - Мониторинг веб-сайтов.
x	integer	Горизонтальная позиция от левой стороны панели. Возможные значения:[1] 0-71.
y	integer	Вертикальная позиция от верхней части панели. Возможные значения:[1] 0-63.
name	string	Пользовательское имя виджета.
width	integer	Ширина виджета. Возможные значения:[1] 1-72.

Element	Type	Description
height	integer	Высота виджета. Возможные значения:[1] 1-64.
hide_header	string	Скрыть заголовок виджета. Возможные значения:[1] NO (0, по умолчанию), YES (1).
fields		Корневой элемент для полей виджета панели.

Note:
См. также: Виджет панели (обратитесь к соответствующему свойству с совпадающим именем).

Поле виджета панели

Element	Type	Description
type	string	(required) Тип поля виджета. Возможные значения:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(required) Имя поля виджета.
value	mixed	(required) Значение поля виджета в зависимости от типа поля.
host	string	Узел сети элемента данных.
key	string	Ключ элемента данных.

Note:
См. также: Поле виджета панели (обратитесь к соответствующему свойству с совпадающим именем).

Сноски

[1] Целочисленные значения API в скобках, например ENABLED (0), приведены только для справки. Для получения дополнительной информации см. связанную страницу объекта API в строке таблицы или в конце каждого раздела.

Регулярные выражения

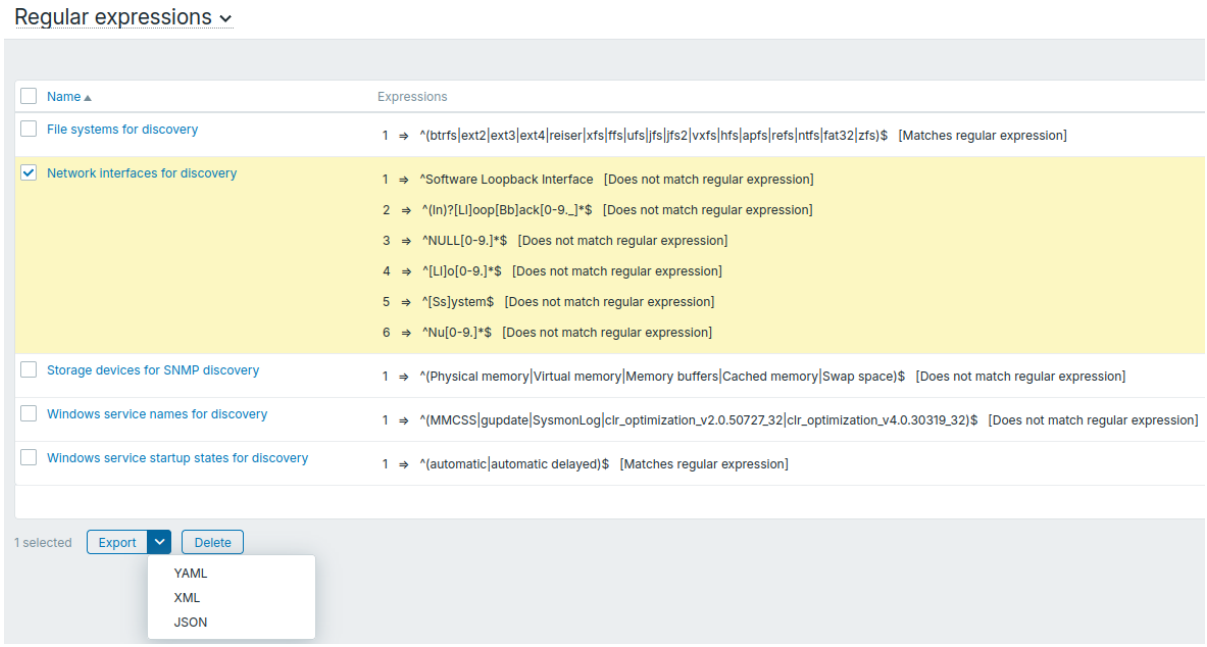
Обзор

Регулярные выражения экспортируются со всеми связанными объектами и связями между объектами.

Экспорт

Чтобы экспортировать регулярные выражения, выполните следующие действия:

1. Перейдите в *Administration* > *General* > *Regular expressions*.
2. Отметьте флажками регулярные выражения, которые нужно экспортировать.
3. Нажмите *Export* под списком, чтобы экспортировать в формате YAML по умолчанию, или нажмите стрелку, чтобы выбрать другой формат экспорта.



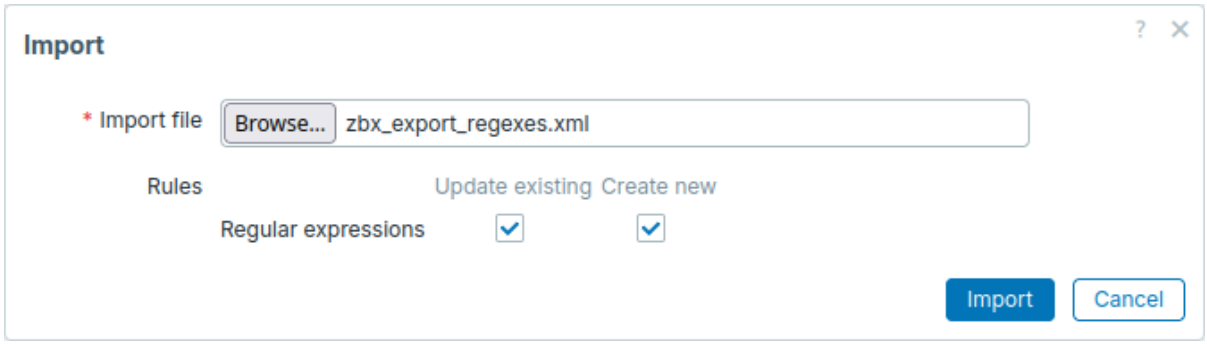
В зависимости от выбранного формата регулярные выражения экспортируются в локальный файл с именем по умолчанию:

- `zbx_export_regexes.yaml` - при экспорте в YAML;
- `zbx_export_regexes.xml` - при экспорте в XML;
- `zbx_export_regexes.json` - при экспорте в JSON.

Импорт

Чтобы импортировать регулярные выражения, выполните следующие действия:

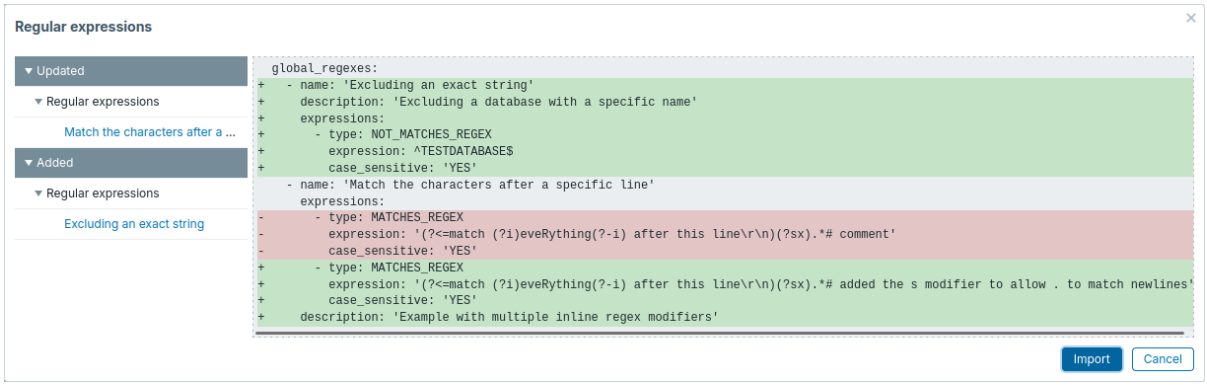
1. Перейдите в раздел *Администрирование* > *Общие* > *Регулярные выражения*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл импорта.
4. Установите флажки правил импорта, которые необходимо применить (см. ниже).
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Правила импорта:

Правило	Описание
Обновить существующие	Существующие регулярные выражения будут обновлены с использованием данных из файла импорта, при этом существующие выражения с тем же именем будут полностью перезаписаны. В противном случае они не будут обновлены.
Создать новые	Новые регулярные выражения будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.

На следующем экране можно просмотреть содержимое импортируемых регулярных выражений. Если это новое регулярное выражение, все данные будут отображены зеленым цветом. При обновлении существующего регулярного выражения новые данные регулярного выражения будут выделены зеленым цветом; удаленные данные будут выделены красным цветом; неизмененные данные будут отображаться на сером фоне.



Меню слева можно использовать для навигации по списку изменений. В разделе *Обновлено* выделены все изменения, внесенные в существующие регулярные выражения. В разделе *Добавлено* перечислены новые регулярные выражения. Нажмите стрелку, чтобы развернуть или свернуть группу регулярных выражений.

Просмотрите изменения регулярных выражений, затем нажмите *Импорт*, чтобы выполнить импорт регулярных выражений. Сообщение об успешном или неуспешном импорте будет отображено в веб-интерфейсе.

Формат экспорта

Экспорт в YAML:

```
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'File systems for discovery'
      description: '                    ,                                                    '
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Storage devices for SNMP discovery'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

Экспортируемые элементы

Экспортируемые элементы регулярных выражений:

Element	Type	Description
global_regexes	object	Корневой элемент для регулярных выражений.
name	string	(required) Имя регулярного выражения.
description	text	Описание регулярного выражения.
expressions	object	Корневой элемент для регулярных выражений.
type	string	Тип регулярного выражения.
expression	string	Регулярное выражение.
exp_delimiter	string	Разделитель регулярного выражения; присутствует, когда тип — *Contains any substring from list*.
case_sensitive	string	Параметр чувствительности к регистру для регулярного выражения.

Note:
См. также: Объект регулярного выражения (см. соответствующее свойство с совпадающим именем).

Регулярные выражения

Обзор Perl Compatible Regular Expressions (PCRE, PCRE2) поддерживаются в Zabbix.

В Zabbix есть два способа использования регулярных выражений:

- Ввод регулярного выражения вручную
- Использование глобального регулярного выражения, созданного в Zabbix

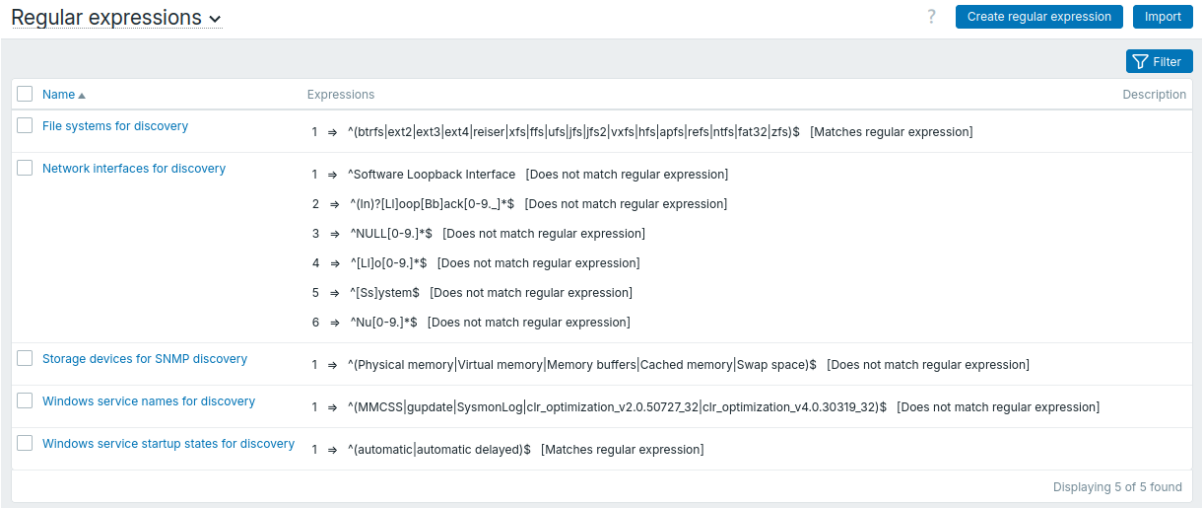
Регулярные выражения Вы можете вручную ввести регулярное выражение в поддерживаемых местах.
 Обратите внимание, что выражение не должно начинаться с @, так как этот символ используется в Zabbix для ссылки на глобальные регулярные выражения.

Attention:
При использовании регулярных выражений возможно исчерпание памяти стека. Подробнее см. в PCRE PERFORMANCE.

Обратите внимание, что при сопоставлении в многострочном режиме якоря ^ и $ соответствуют началу и концу каждой строки соответственно, а не началу и концу всей строки.

Также см. примеры правильного экранирования в различных контекстах.

Глобальные регулярные выражения Zabbix предоставляет в веб-интерфейсе расширенный редактор для создания и тестирования сложных регулярных выражений.
 Чтобы открыть его, перейдите в *Administration* > *General* > *Regular expressions*.
 Zabbix включает ряд предопределенных глобальных регулярных выражений в наборе данных по умолчанию.

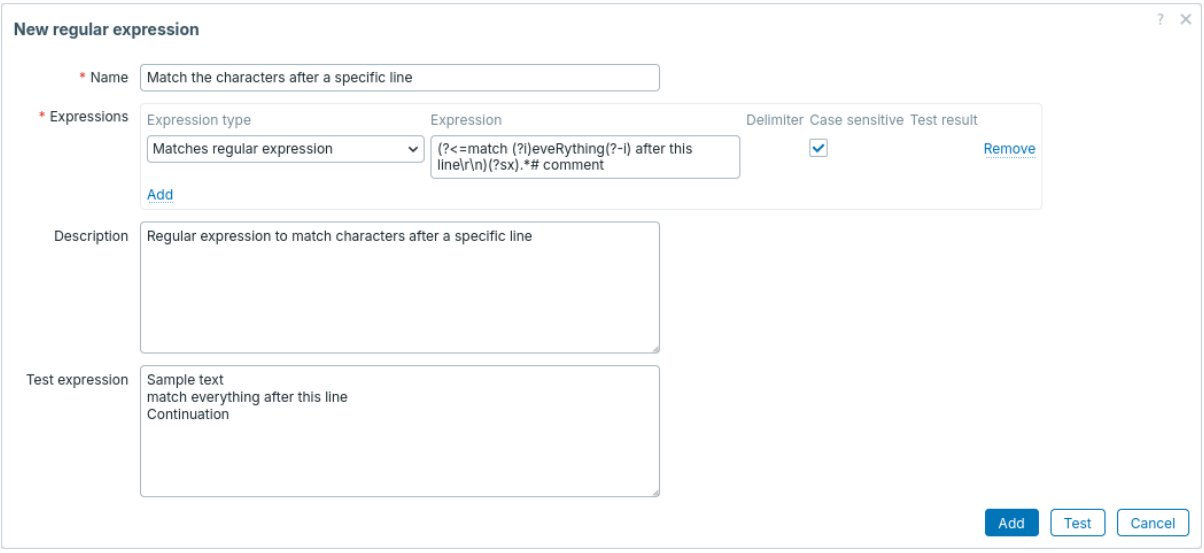


Вы можете фильтровать список регулярных выражений по имени и описанию, чтобы найти нужные выражения.

Настройка

Чтобы создать глобальное регулярное выражение:

1. Нажмите *Создать регулярное выражение*.
2. Введите параметры регулярного выражения в форме.



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя	Укажите имя регулярного выражения. Разрешены любые символы Unicode.
Выражения	Нажмите *Добавить* в разделе *Выражения*, чтобы добавить новое подвыражение.
Тип выражения	Выберите тип выражения: **Содержит строку** - соответствует подстроке. **Содержит любую подстроку из списка** - соответствует любой подстроке из списка с разделителями. В качестве разделителя можно использовать запятую `,`, точку `.` или косую черту `/`. **Не содержит строку** - соответствует любой строке, кроме подстроки. **Соответствует регулярному выражению** - соответствует регулярному выражению. **Не соответствует регулярному выражению** - не соответствует регулярному выражению.
Выражение	Введите подстроку или регулярное выражение. Максимальная длина: 2048 символов.
Разделитель	Запятая `,`, точка `.` или косая черта `/` для разделения текстовых строк в регулярном выражении. Этот параметр активен только при выборе типа выражения *Содержит любую подстроку из списка*.
С учетом регистра	Флажок, указывающий, должно ли регулярное выражение учитывать регистр букв.
Описание	Введите описание регулярного выражения.
Проверка выражения	Значение, используемое для проверки регулярного выражения.

Косая черта `/` в выражении обрабатывается буквально, а не как разделитель. Это позволяет сохранять выражения, содержащие косую черту, без ошибок.

Attention:
Имя пользовательского регулярного выражения в Zabbix может содержать запятые, пробелы и другие символы. Если это может привести к неправильной интерпретации при ссылке на выражение (например, запятая в параметре ключа элемента данных), всю ссылку можно заключить в кавычки, например: *"@My custom regexp for purpose1, purpose2"*.
 В других местах имена регулярных выражений не следует заключать в кавычки (например, в свойствах правила LLD).

После создания регулярного выражения его можно использовать в нескольких местах веб-интерфейса, ссылаясь на его имя с префиксом `@`, например `@mycustomregexp`.

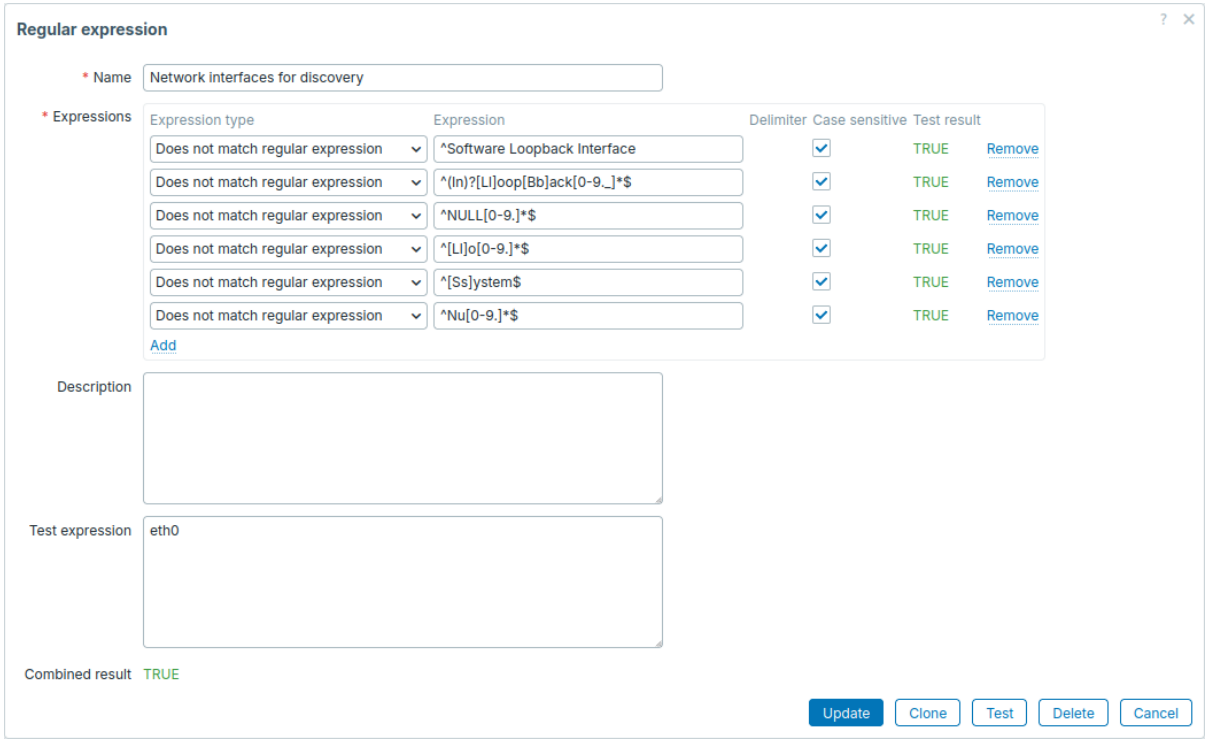
Кнопки формы

Кнопки в нижней части формы позволяют выполнять различные операции:

Button	Description
Add	Добавить регулярное выражение. Эта кнопка доступна только для новых регулярных выражений.
Update	Обновить свойства регулярного выражения.
Clone	Создать еще одно регулярное выражение на основе свойств текущего регулярного выражения.
Test	Проверить регулярное выражение и его подвыражения на предоставленном значении.
Delete	Удалить регулярное выражение.
Cancel	Отменить редактирование свойств регулярного выражения.

Проверка регулярного выражения

Вы можете проверить регулярное выражение и его подвыражения, введя значение в поле *Test expression* и нажав кнопку *Test* button.



В столбце *Test result* отображается статус каждого подвыражения. Поле *Combined result* отображает общий статус регулярного выражения. Если определено несколько подвыражений, Zabbix использует логический оператор AND для вычисления *Combined result*; то есть, если хотя бы один результат равен `False`, *Combined result* также будет `False`.

Регулярные выражения по умолчанию для глобальных правил

Набор данных по умолчанию включает следующие предопределенные глобальные регулярные выражения:

Name	Expression	Matches
File systems for discovery	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	”btrfs” or ”ext2” or ”ext3” or ”ext4” or ”reiser” or ”xfs” or ”ffs” or ”ufs” or ”jfs” or ”jfs2” or ”vxfs” or ”hfs” or ”apfs” or ”refs” or ”ntfs” or ”fat32” or ”zfs”.
Network interfaces for discovery	`^Software Loopback Interface`	Строки, начинающиеся с ”Software Loopback Interface”.
	`^lo$`	”lo”.
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Строки, при необходимости начинающиеся с ”In”, затем с ”L” или ”l”, затем с ”oop”, затем с ”B” или ”b”, затем с ”ack” и, при необходимости, далее с любым количеством цифр, точек или символов подчеркивания.
	`^NULL[0-9.]*$`	Строки, начинающиеся с ”NULL” и, при необходимости, далее с любым количеством цифр или точек.
	`^[Ll]o[0-9.]*$`	Строки, начинающиеся с ”Lo” или ”lo” и, при необходимости, далее с любым количеством цифр или точек.
	`^[Ss]ystem$`	”System” или ”system”.
	`^Nu[0-9.]*$`	Строки, начинающиеся с ”Nu” и, при необходимости, далее с любым количеством цифр или точек.
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” or ”Virtual memory” or ”Memory buffers” or ”Cached memory” or ”Swap space”.
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS” or ”gupdate” or ”SysmonLog” or matching ”clr_optimization_v2.0.50727_32” or ”clr_optimization_v4.0.30319_32”, where each dot can be replaced by any character except a newline.

Name	Expression	Matches
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	”automatic” or ”automatic delayed”.

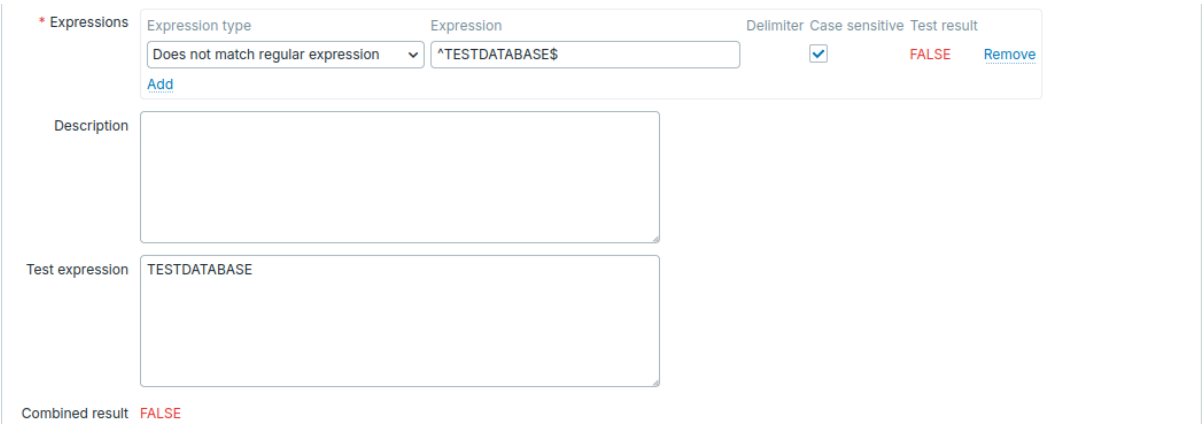
Импорт/экспорт регулярного выражения

Вы можете импортировать или экспортировать глобальные регулярные выражения, как описано на странице настройки экспорта/импорта Regular expressions.

Примеры Пример исключения точной строки

Используйте следующее выражение в низкоуровневом обнаружении, чтобы обнаруживать базы данных, кроме базы данных с определенным именем:

```
^TESTDATABASE$
```

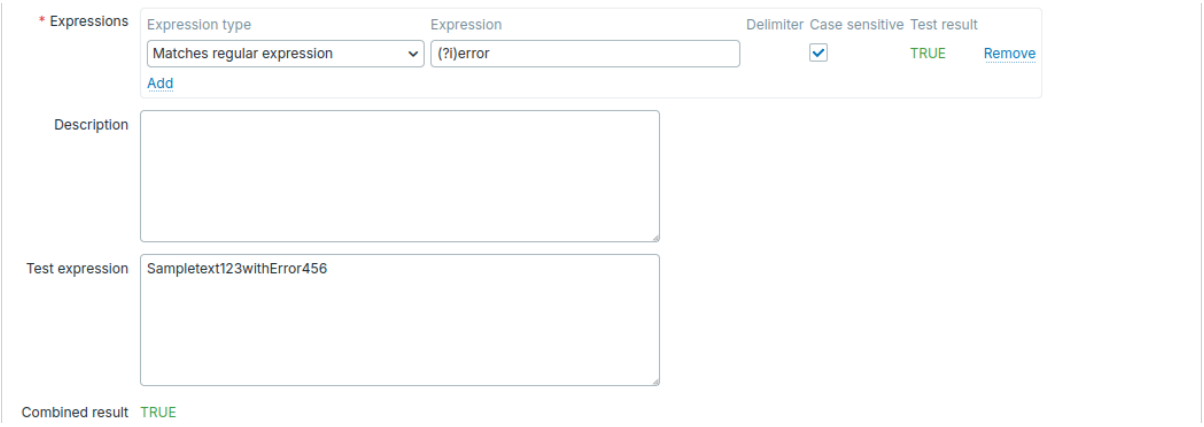


И *Test result*, и *Combined result* отображают `FALSE`: имя совпадает с ”TESTDATABASE”, поэтому условие *Does not match regular expression* не выполняется.

Пример с встроенным модификатором regex для нечувствительного к регистру совпадения

Используйте следующее регулярное выражение, которое включает встроенный модификатор `(?i)` для выполнения нечувствительного к регистру совпадения символов ”error”:

```
(?i)error
```



И *Test result*, и *Combined result* отображают `TRUE`: символы ”error” совпадают даже при выбранном *Case sensitive*, потому что встроенный модификатор `(?i)` делает совпадение нечувствительным к регистру.

Пример с несколькими встроенными модификаторами регулярных выражений

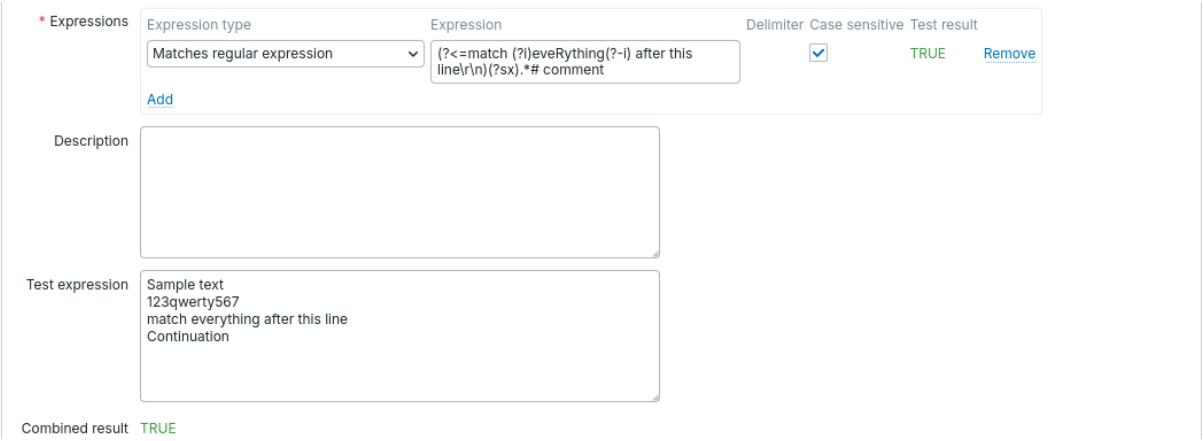
Используйте следующее регулярное выражение, которое включает несколько встроенных модификаторов, чтобы сопоставить символы после определенной строки:

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

где:

- `(?i)` - включает нечувствительное к регистру сопоставление;

- `(?-i)` - отключает нечувствительное к регистру сопоставление;
- `(?s)` - включает режим dotall, позволяя `.` сопоставляться с символами новой строки;
- `(?x)` - включает расширенный режим, позволяя игнорировать пробелы и комментарии в регулярном выражении.



И *Test result*, и *Combined result* отображают `TRUE`: сопоставляются символы, следующие за указанной строкой.

Attention:
Модификатор g нельзя указывать встроенно. Список доступных модификаторов можно найти на странице man pcresyntax. Для получения дополнительной информации о синтаксисе PCRE см. документацию PCRE в формате HTML.

Поддержка регулярных выражений по расположению

Расположение		Регулярное выражение	Глобальное регулярное выражение	Многострочное сопоставление	Комментарии
Элементы данных агента					
	eventlog[]	Да	Да	Да	параметры `regexp`, `severity`, `source`, `eventid`
	eventlog.count[]				параметры `regexp`, `severity`, `source`, `eventid`
	log[]				параметр `regexp`
	log.count[]				
	logrt[]		Да/Нет		параметр `regexp` поддерживает оба варианта, параметр `file_regexp` поддерживает только неглобальные выражения
	logrt.count[]				
	proc.cpu.util[]		Нет	Нет	параметр `cmdline`
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				параметры `device` и `sensor` в Linux 2.4
	system.hw.macaddr[]				параметр `interface`
	system.sw.packages[]				параметр `regexp`
	system.sw.packages.get[]				параметр `regexp`
	vfs.dir.count[]				параметры `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.dir.get[]				параметры `regex_incl`, `regex_excl`, `regex_excl_dir`

Расположение		Регулярное выражение	Глобальное регулярное выражение	Многострочное сопоставление	Комментарии
	vfs.dir.size[]				параметры `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.file.regexp[] vfs.file.regmatch[] web.page.regexp[]			Да	параметр `regexp`
SNMP traps					
	snmptrap[]	Да	Да	Нет	параметр `regexp`
Предобработка значения элемента данных		Да	Нет	Нет	параметр `pattern`
Функции для триггеров/вычисляемых элементов данных					
	count()	Да	Да	Да	параметр `pattern`, если параметр `operator` имеет значение *regexp* или *iregexp*
	countunique()	Да	Да		
	find()	Да	Да		
	logeventid()	Да	Да	Нет	параметр `pattern`
	logsource()				
Низкоуровневое обнаружение					
	Фильтры	Да	Да	Нет	поле *Regular expression*
	Переопределения	Да	Нет		в условиях *matches*, *does not match* для операций *Operation*
Условия действий		Да	Нет	Нет	в условиях автрорегистрации *matches*, *does not match* для *Host name* и *Host metadata*
Скрипты		Да	Да	Нет	поле *Input validation rule*
Веб-мониторинг		Да	Нет	Да	*Variables* с префиксом **regex:** поле *Required string*
Контекст пользовательского макроса		Да	Нет	Нет	в контексте макроса с префиксом **regex:**
Функции макросов					
	regsub() iregsub()	Да	Нет	Нет	параметр `pattern`
Индикаторы ссылок на картах		Да	Нет	Нет	поле *Pattern* (для текстовых элементов данных)
Сопоставление значков		Да	Да	Нет	поле *Expression*
Сопоставление значений		Да	Нет	Нет	поле *Value*, если тип сопоставления — `regexp`
Параметры конфигурации агента					
	AllowKeyRegexp DenyKeyRegexp	Да	Нет	Нет	Шаблон

События и разрешение

Обзор В Zabbix генерируются несколько типов событий:

- события триггеров - всякий раз, когда триггер изменяет свой статус (*OK→PROBLEM→OK*)
- события сервисов - всякий раз, когда сервис изменяет свой статус (*OK→PROBLEM→OK*)
- события обнаружения - когда обнаруживаются узлы сети или сервисы
- события авто-регистрации - когда активные агенты автоматически регистрируются сервером
- внутренние события - когда элемент данных/правило низкоуровневого обнаружения становится неподдерживаемым или триггер переходит в неизвестное состояние

События имеют временную метку и могут служить основой для таких действий, как отправка уведомления по электронной почте, создание тикетов или выполнение удаленных команд.

Чтобы просмотреть подробности события в веб-интерфейсе, перейдите в *Monitoring* > *Problems*. Там можно щелкнуть по дате и времени события, чтобы открыть страницу с подробностями события, где можно увидеть дополнительную информацию о событии, включая то, было ли оно подавлено из-за обслуживания или подавлено вручную пользователем.

Разрешение проблем События проблем, созданные триггерами, автоматически разрешаются, если условие, вызвавшее проблему, больше не существует.

Чтобы добавить гибкость и удобство в управление событиями проблем, доступны следующие возможности:

- Корреляция событий
- Подтверждение и подавление событий
- Ручное закрытие проблем
- Периоды обслуживания - определяют периоды, когда проблемы подавляются
- Тегирование - помечать проблемы по общему признаку

События триггера Изменение состояния триггера является наиболее частым и наиболее важным источником событий. Каждый раз, когда триггер меняет свое состояние, создается событие. Событие содержит сведения об изменении состояния триггера — когда это произошло и каково новое состояние.

Триггерами создаются два типа событий — `Problem` и `OK`.

События проблемы

Событие проблемы создается:

- когда выражение триггера принимает значение TRUE, если триггер находится в состоянии OK;
- каждый раз, когда выражение триггера принимает значение TRUE, если для триггера включена генерация нескольких событий проблемы.

OK-события

OK-событие закрывает связанные с ним problem-события и может быть создано 3 компонентами:

- триггеры - на основе настроек ’OK event generation’ и ’OK event closes’;
- корреляция событий
- диспетчер задач - когда событие закрывается вручную

Триггеры имеют настройку «Генерация события OK», которая определяет, как генерируются события OK:

- *Выражение* - событие OK генерируется для триггера в состоянии проблемы, когда его выражение вычисляется как FALSE. Это самый простой вариант, включенный по умолчанию.
- *Выражение восстановления* - событие OK генерируется для триггера в состоянии проблемы, когда его выражение вычисляется как FALSE, а выражение восстановления вычисляется как TRUE. Это можно использовать, если критерии восстановления триггера отличаются от критериев проблемы.
- *Нет* - событие OK никогда не генерируется. Это можно использовать совместно с множественной генерацией событий проблемы, чтобы просто отправлять уведомление, когда что-то происходит.

Кроме того, у триггеров есть настройка «Событие OK закрывает», которая определяет, какие события проблемы будут закрыты:

- *Все проблемы* - событие OK закроет все открытые проблемы, созданные триггером
- *Все проблемы, если значения тегов совпадают* - событие OK закроет открытые проблемы, созданные триггером и имеющие хотя бы одно совпадающее значение тега. Тег определяется настройкой триггера «Тег для

сопоставления». Если нет событий проблемы, которые нужно закрыть, событие OK не генерируется. Это часто называется корреляцией событий на уровне триггера.

Корреляция событий

Корреляция событий (также называемая глобальной корреляцией событий) — это способ настройки пользовательских правил закрытия событий (в результате чего создаются события со статусом «OK»).

Правила определяют, как новые события проблем сопоставляются с существующими событиями проблем, позволяя закрывать новое событие или сопоставленные события путем создания соответствующих событий со статусом «OK».

Однако корреляцию событий необходимо настраивать очень тщательно, поскольку она может негативно повлиять на производительность обработки событий или, при неправильной настройке, закрыть больше событий, чем предполагалось (в худшем случае могут быть закрыты все события проблем). Несколько советов по настройке:

1. всегда уменьшайте область корреляции, задавая уникальный тег для управляющего события (события, сопоставляемого со старыми событиями), и используйте условие корреляции «тег нового события»
2. не забывайте добавлять условие на основе старого события при использовании операции «закрыть старое событие», иначе могут быть закрыты все существующие проблемы
3. избегайте использования общих имен тегов, применяемых в разных конфигурациях корреляции

Диспетчер задач

Если для триггера включена настройка «Разрешить ручное закрытие», то можно вручную закрывать события проблем, сгенерированные триггером. Это делается во веб-интерфейсе при обновлении проблемы. Событие не закрывается напрямую — вместо этого создается задача «закрыть событие», которая вскоре обрабатывается диспетчером задач. Диспетчер задач сгенерирует соответствующее событие OK, и событие проблемы будет закрыто.

События сервиса События сервиса создаются только в том случае, если для этих событий включены действия сервиса. В этом случае каждое изменение состояния сервиса создает новое событие:

- Событие проблемы - когда состояние сервиса изменяется с `OK` на `Problem`
- Событие OK - когда состояние сервиса изменяется с `Problem` на `OK`

Событие содержит сведения об изменении состояния сервиса - когда оно произошло и каково новое состояние.

События обнаружения Zabbix периодически сканирует диапазоны IP-адресов, определенные в правилах сетевого обнаружения. Частота проверки настраивается отдельно для каждого правила. После обнаружения узла сети или службы генерируется событие обнаружения (или несколько событий).

Zabbix генерирует следующие события:

Событие	Когда генерируется
Служба доступна	Каждый раз, когда Zabbix обнаруживает активную службу.
Служба недоступна	Каждый раз, когда Zabbix не может обнаружить службу.
Узел сети доступен	Если хотя бы одна из служб для IP-адреса доступна.
Узел сети недоступен	Если ни одна из служб не отвечает.
Служба обнаружена	Если служба снова доступна после простоя или обнаружена впервые.
Служба потеряна	Если служба потеряна после того, как была доступна.
Узел сети обнаружен	Если узел сети снова доступен после простоя или обнаружен впервые.
Узел сети потерян	Если узел сети потерян после того, как был доступен.

События авторегистрации активного агента Автрегистрация активного агента создает события в Zabbix.

Если настроена, событие автрегистрации активного агента создается, когда ранее неизвестный активный агент запрашивает проверки или если метаданные узла сети изменились. Сервер добавляет новый автоматически зарегистрированный узел сети, используя полученные IP-адрес и порт агента.

Дополнительные сведения см. на странице авторегистрации активного агента.

Внутренние события Внутренние события происходят, когда:

- элемент данных меняет состояние с ’normal’ на ’unsupported’
- элемент данных меняет состояние с ’unsupported’ на ’normal’
- правило обнаружения низкого уровня меняет состояние с ’normal’ на ’unsupported’
- правило обнаружения низкого уровня меняет состояние с ’unsupported’ на ’normal’
- триггер меняет состояние с ’normal’ на ’unknown’
- триггер меняет состояние с ’unknown’ на ’normal’

Цель введения внутренних событий — дать пользователям возможность получать уведомления при возникновении любого внутреннего события, например, когда элемент данных становится неподдерживаемым и перестает собирать данные.

Внутренние события создаются только тогда, когда для этих событий включены внутренние действия. Чтобы остановить генерацию внутренних событий (например, для элементов данных, которые становятся неподдерживаемыми), отключите все действия для внутренних событий в *Alerts* > *Actions* > *Internal actions*.

Note:
Если внутренние действия отключены, то пока объект находится в состоянии 'unsupported', для этого объекта все равно будет создано событие восстановления.
Если внутренние действия включены, то пока объект находится в состоянии 'unsupported', для этого объекта будет создано событие восстановления, даже если для объекта не было создано 'problem event'.

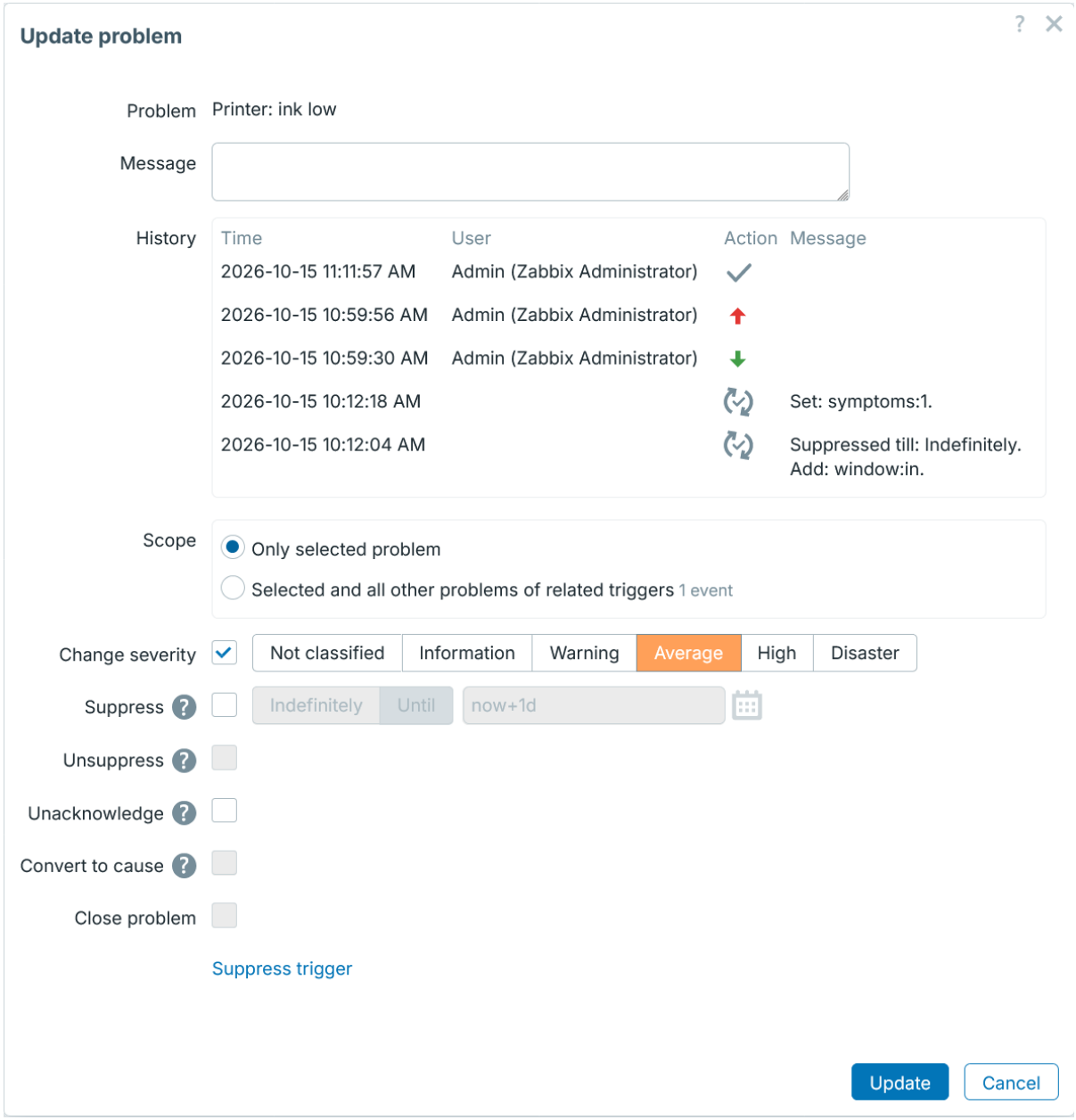
См. также: Получение уведомления о неподдерживаемых элементах данных

Подтверждение и подавление проблем

Проблемные события в Zabbix могут быть подтверждены или подавлены пользователями.

Обновление проблем Всплывающее окно обновления проблемы позволяет:

- Добавить комментарий к проблеме.
- Просмотреть комментарии и действия, выполненные до текущего момента.
- Изменить важность проблемы.
- Подавить проблему или отменить ее подавление.
- Подтвердить проблему или отменить ее подтверждение.
- Преобразовать проблему-симптом в проблему-причину.
- Вручную закрыть проблему.
- Создать внеплановое обслуживание для триггера.



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Problem	Если выбрана только одна проблема, отображается имя проблемы. Если выбрано несколько проблем, отображается *N problems selected*.
Message	Введите текст комментария к проблеме (не более 2048 символов).
History	Отображаются предыдущие действия и комментарии к проблеме, а также время и сведения о пользователе. Значение значков, используемых для обозначения действий пользователей, см. на странице сведений о событии. Обратите внимание, что история отображается только в том случае, если для обновления выбрана одна проблема.
Scope	Определите область действия таких операций, как изменение важности, подтверждение или ручное закрытие проблем: **Only selected problem** - затронет только это событие **Selected and all other problems of related triggers** - при подтверждении или закрытии проблемы затронет это событие и все остальные проблемы, которые до текущего момента не были подтверждены или закрыты. Если область действия включает уже подтвержденные или закрытые проблемы, повторно подтверждать или закрывать их не будут. При этом количество операций добавления сообщения и изменения важности не ограничено.

Параметр	Описание
Change severity	Установите флажок и нажмите кнопку важности, чтобы обновить важность проблемы. Флажок изменения важности доступен, если для хотя бы одной из выбранных проблем имеются разрешения на чтение и запись. При нажатии *Update* будут обновлены только проблемы, для которых доступны такие разрешения. Если ни для одного из выбранных триггеров нет разрешений на чтение и запись, флажок отключен.
Suppress	Установите флажок, чтобы подавить проблему: **Indefinitely** - подавить на неопределенный срок **Until** - подавить до указанного времени. Поддерживаются абсолютный и относительный форматы времени, например: `now+1d` - на один день с текущего момента (по умолчанию) `now/w` - до конца текущей недели `2022-05-28 12:00:00` - до указанной даты и времени Обратите внимание, что простой период (например, `1d`, `1w`) не поддерживается. Доступность этого параметра зависит от настроек пользовательской роли "Suppress problems". См. также: Подавление проблем
Unsuppress	Установите флажок, чтобы отменить подавление проблемы. Этот флажок активен только в том случае, если подавлена хотя бы одна из выбранных проблем. Доступность этого параметра зависит от настроек пользовательской роли "Suppress problems".
Acknowledge	Установите флажок, чтобы подтвердить проблему. Этот флажок доступен, если среди выбранных есть хотя бы одна неподтвержденная проблема. Невозможно добавить еще одно подтверждение для уже подтвержденной проблемы (однако можно добавить еще один комментарий).
Unacknowledge	Установите флажок, чтобы отменить подтверждение проблемы. Этот флажок доступен, если среди выбранных есть хотя бы одна подтвержденная проблема.
Convert to cause	Установите флажок, чтобы преобразовать проблемы-симптомы в проблемы-причины.
Close problem	Установите флажок, чтобы вручную закрыть выбранные проблемы. Флажок закрытия проблемы доступен, если параметр *Allow manual close* установлен в конфигурации триггера хотя бы для одной из выбранных проблем. При нажатии *Update* будут закрыты только проблемы, для которых разрешено закрытие. Если ни одну проблему нельзя закрыть вручную, флажок отключен. Уже закрытые проблемы не будут закрываться повторно.
Suppress trigger	Нажмите, чтобы начать внеплановое обслуживание для триггера. Откроется форма настройки периода обслуживания. В ней будет предварительно выбран однократный период обслуживания продолжительностью 1 ч, начинающийся с текущего момента (продолжительность можно настроить в настройках и профиле пользователя). Соответствующий триггер также будет предварительно выбран. Имя обслуживания будет предварительно заполнено в формате `Ad-hoc: <event name>`.

Подтверждение проблемы Если пользователь получает уведомление о событии проблемы, он может перейти в веб-интерфейс Zabbix, открыть всплывающее окно обновления проблемы для этой проблемы одним из способов, перечисленных ниже, и подтвердить проблему. При подтверждении он может добавить комментарий, например указать, что работает над ней, или написать что-либо еще по своему усмотрению.

Таким образом, если другой пользователь системы обнаружит ту же проблему, он сразу увидит, была ли она подтверждена, а также все комментарии на данный момент.

Таким образом, процесс устранения проблем, в котором участвует более одного пользователя системы, может выполняться согласованно.

Статус подтверждения также используется при определении операций действий. Например, можно задать, чтобы уведомление отправлялось менеджеру более высокого уровня только в том случае, если событие не было подтверждено в течение некоторого времени.

Чтобы подтверждать события и добавлять к ним комментарии, пользователь должен иметь как минимум права на чтение соответствующих триггеров. Чтобы изменить серьезность проблемы или закрыть проблему, пользователь должен иметь права на чтение и запись для соответствующих триггеров.

Существует **несколько** способов открыть всплывающее окно обновления проблемы, которое позволяет подтвердить

проблему.

- Можно выбрать проблемы в *Monitoring* > *Problems*, а затем нажать *Mass update* под списком
- Можно нажать *Update* в столбце *Update* у проблемы в:
  - *Dashboards* (виджеты *Problems* и *Problems by severity*)
  - *Monitoring* > *Problems*
  - *Monitoring* > *Problems* > *Event details*
- Можно нажать на ячейку нерешенной проблемы в:
  - *Dashboards* (виджет *Trigger overview*)

Всплывающее меню содержит пункт *Update*, который перенаправит вас в окно обновления проблемы.

На основе информации о подтверждении можно настроить отображение количества проблем на панели мониторинга или картах. Для этого необходимо выбрать значение параметра *Отображение проблем*, доступного как в настройках карты, так и в виджете панели мониторинга *Проблемы по степени серьезности*.

Можно отображать общее количество проблем, количество неподтвержденных проблем отдельно от общего количества или только количество неподтвержденных проблем.

На основе информации об обновлении проблемы (подтверждение и т. д.) можно настроить операции обновления: отправку сообщения или выполнение удаленных команд.

Подавление проблем Подавление проблем позволяет временно скрыть проблему, чтобы заняться ею позже. Это удобно для очистки списка проблем и выделения наиболее высокого приоритета для самых срочных вопросов. Например, иногда в выходные может возникнуть проблема, которая не настолько срочна, чтобы заниматься ею немедленно, поэтому ее можно «отложить» до утра понедельника.

Подавление проблем позволяет скрыть *одну* проблему, в отличие от подавления с помощью обслуживания узла сети, при котором скрываются все проблемы узла сети, находящегося на обслуживании.

Операции для действий триггеров будут приостановлены для подавленных проблем так же, как это происходит при обслуживании узла сети.

Проблема может быть подавлена через окно **обновления проблемы**, где подавление является одной из опций обновления проблемы наряду с добавлением комментария, изменением серьезности, подтверждением и т. д.

Проблема также может быть снята с подавления через то же окно обновления проблемы.

Отображение

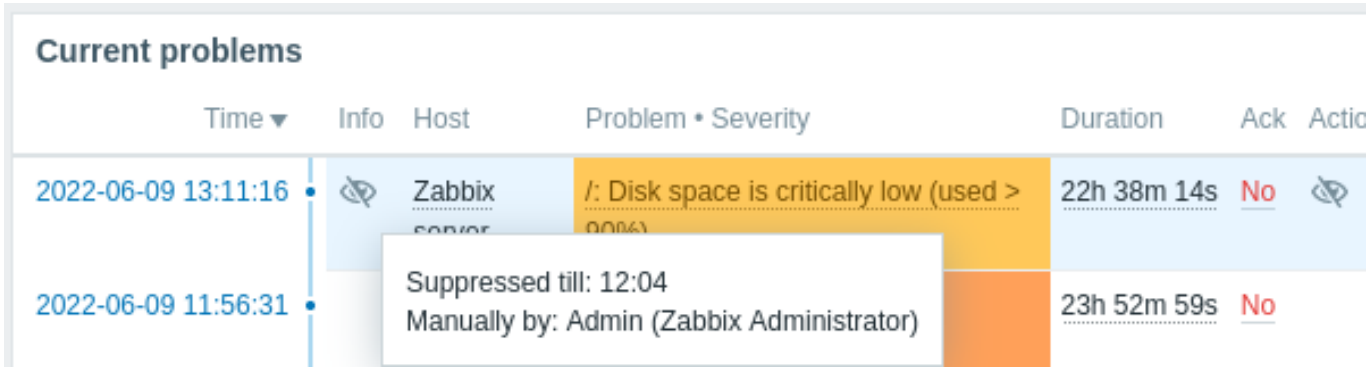
После подавления проблема помечается мигающим значком подавления  в столбце *Info* перед тем, как быть скрытой.

Если подавление связано с обслуживанием узла сети, значок подавления в столбце *Info* отображается оранжевым цветом. Если применяются и обслуживание, и ручное подавление, приоритет в цвете значка имеет ручное подавление.

Значок подавления мигает, пока задача подавления находится в списке ожидания. После того как диспетчер задач подавит проблему, значок перестанет мигать. Если значок подавления продолжает мигать в течение длительного времени, это может указывать на проблему с сервером, например если сервер недоступен и диспетчер задач не может завершить задачу. Та же логика применяется и к снятию подавления. В короткий период после отправки задачи и до ее выполнения сервером значок снятия подавления мигает.

Подавленная проблема может быть как скрыта, так и показана, в зависимости от настроек фильтра проблемы/виджета.

При отображении в списке проблем подавленная проблема помечается значком подавления, а сведения о подавлении показываются при наведении курсора:



Сведения о подавлении также отображаются во всплывающем окне при наведении курсора на значок подавления в столбце *Actions*.

Ручное закрытие проблем

Обзор Хотя обычно события проблемы автоматически разрешаются, когда состояние триггера переходит из `Problem` в `OK`, бывают случаи, когда трудно определить, была ли проблема устранена с помощью выражения триггера. В таких случаях проблему необходимо закрыть вручную.

Например, *syslog* может сообщать, что некоторые параметры ядра нужно настроить для оптимальной производительности. В этом случае информация о проблеме передается администраторам Linux, они исправляют ее, а затем закрывают проблему вручную.

Проблемы можно закрывать вручную только для триггеров, у которых включена опция *Allow manual close*.

Когда проблема "закрывается вручную", Zabbix создает новую внутреннюю задачу для сервера Zabbix. Затем процесс *task manager* выполняет эту задачу и генерирует событие OK, тем самым закрывая событие проблемы.

Сгенерированное событие *OK* включает полный набор тегов события, разрешенных для этого события (включая теги, унаследованные от шаблонов, узлов сети и триггеров). Эти теги доступны в уведомлениях и в макросах, таких как `{EVENT.RECOVERY.TAGS}` и `{EVENT.RECOVERY.TAGSJSON}`.

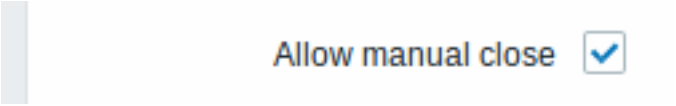
Ручное закрытие проблемы не означает, что соответствующий триггер больше никогда не перейдет в состояние *Problem*. Выражение триггера повторно оценивается и может снова привести к проблеме:

- Когда поступают новые данные для любого элемента данных, включенного в выражение триггера (обратите внимание, что значения, отброшенные на этапе предварительной обработки с ограничением частоты, не считаются полученными и не вызовут повторную оценку выражения триггера);
- Когда в выражении используются функции date and time и/или nodata().

Конфигурация Для ручного закрытия проблемы требуется два шага.

Конфигурация триггера

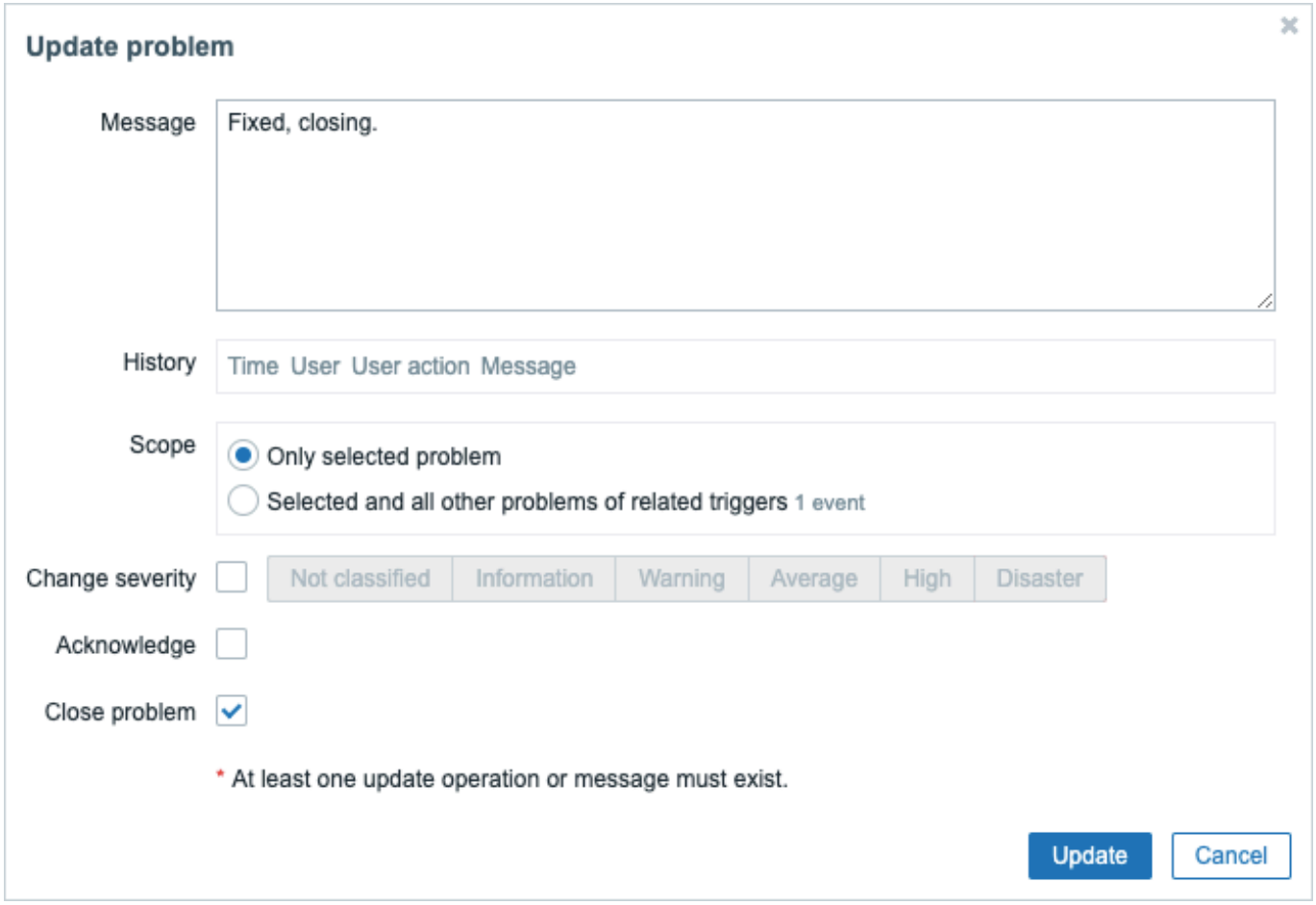
В конфигурации триггера включите параметр *Разрешить ручное закрытие*.



Окно обновления проблемы

Если для триггера с флагом *Manual close* возникает проблема, вы можете открыть всплывающее окно обновления проблемы для этой проблемы и закрыть проблему вручную.

Чтобы закрыть проблему, установите флажок *Close problem* в форме и нажмите *Update*.



Все обязательные поля ввода отмечены красной звездочкой.

Запрос обрабатывается сервером Zabbix. Обычно для закрытия проблемы требуется несколько секунд. В течение этого процесса в *Monitoring* > *Problems* в качестве статуса проблемы отображается *CLOSING*.

Проверка Можно проверить, что проблема была закрыта вручную:

- в деталях события, доступных через *Monitoring* > *Problems*;
- с помощью макроса `{EVENT.UPDATE.HISTORY}` в уведомлениях, который предоставит эту информацию.

Периоды обслуживания

Обзор

Техническое обслуживание используется для подавления проблем в заранее определенные периоды времени.

Периоды технического обслуживания можно определить для следующих сущностей:

- Группы узлов сети
- Узлы сети
- Триггеры
- Имена событий
- Теги

Если техническое обслуживание определено для определенных тегов или имен событий, оно будет активировано только для соответствующих триггеров; все остальные триггеры узла сети или группы узлов сети не будут находиться на техническом обслуживании.

Существует два типа технического обслуживания: *Со сбором данных* и *Без сбора данных*.

Во время технического обслуживания типа *Со сбором данных* триггеры обрабатываются как обычно, а события создаются при необходимости. Однако если в настройках действия установлен флажок *Приостановить операции для подавленных проблем*, эскалации проблем приостанавливаются для сущностей, находящихся на техническом обслуживании. В этом случае шаги эскалации, которые могут включать отправку уведомлений или выполнение удаленных команд, игнорируются на протяжении всего периода технического обслуживания. Обратите внимание, что восстановление

проблем и операции обновления во время технического обслуживания не подавляются, подавляются только эскалации. Уведомления о восстановлении не отправляются, если проблема возникла во время технического обслуживания.

Например, если шаги эскалации запланированы через 0, 30 и 60 минут после возникновения проблемы, а через 10 минут после возникновения реальной проблемы начинается техническое обслуживание продолжительностью полчаса, которое заканчивается через 40 минут, второй и третий шаги будут выполнены на полчаса позже, то есть через 60 и 90 минут (при условии, что проблема все еще существует). Аналогично, если проблема возникает во время технического обслуживания, эскалация начнется после его окончания.

Чтобы получать уведомления о проблемах во время технического обслуживания без задержки, необходимо снять флажок *Приостановить операции для подавленных проблем* в настройках действия.

Note:
Если хотя бы один узел сети (используемый в выражении триггера) не находится на техническом обслуживании, Zabbix отправит уведомление о проблеме.

Сервер Zabbix должен работать во время технического обслуживания. Периоды технического обслуживания пересчитываются каждую минуту или сразу после перезагрузки кэша конфигурации, если в период технического обслуживания были внесены изменения.

Процессы таймера проверяют, необходимо ли изменить статус узла сети на «техническое обслуживание» или обратно, в 0 секунд каждой минуты. Кроме того, каждую секунду процесс таймера проверяет, необходимо ли запустить или остановить техническое обслуживание на основе изменений в [периодах технического обслуживания] после обновления конфигурации. Таким образом, скорость запуска и остановки периодов технического обслуживания зависит от интервала обновления конфигурации (по умолчанию 10 секунд). Обратите внимание, что изменения периодов технического обслуживания не включают настройки *Активно с/Активно до*. Кроме того, если сущность (группа узлов сети/узел сети/триггер и т. д.) добавлена в уже активный период технического обслуживания, изменения будут активированы процессом таймера только в начале следующей минуты.

Обратите внимание, что когда узел сети переходит на техническое обслуживание, процессы таймера сервера Zabbix считывают все открытые проблемы, чтобы проверить, требуется ли их подавление. Это может повлиять на производительность, если существует много открытых проблем. Сервер Zabbix также считывает все открытые проблемы при запуске, даже если в этот момент не настроено ни одного периода технического обслуживания.

Обратите внимание, что сервер Zabbix (или прокси) всегда собирает данные независимо от типа технического обслуживания (включая техническое обслуживание типа *Без сбора данных*). Позже сервер игнорирует эти данные, если установлен тип *Без сбора данных*.

После окончания технического обслуживания типа *Без сбора данных* триггеры, использующие функцию `nodata()`, не сработают до следующей проверки в течение периода, который они проверяют.

Если элемент данных журнала добавлен, пока узел сети находится на техническом обслуживании, а затем техническое обслуживание заканчивается, будут собираться только новые записи файла журнала, появившиеся после окончания технического обслуживания.

Если для узла сети, находящегося на техническом обслуживании типа *Без сбора данных*, отправляется значение с временной меткой (например, с помощью Zabbix sender), это значение будет отброшено; однако можно отправить значение с временной меткой за истекший период технического обслуживания, и оно будет принято.

Если пользователь изменяет период технического обслуживания или параметры технического обслуживания (группу узлов сети, узел сети, триггер и т. д.), изменения вступят в силу только после синхронизации кэша конфигурации.

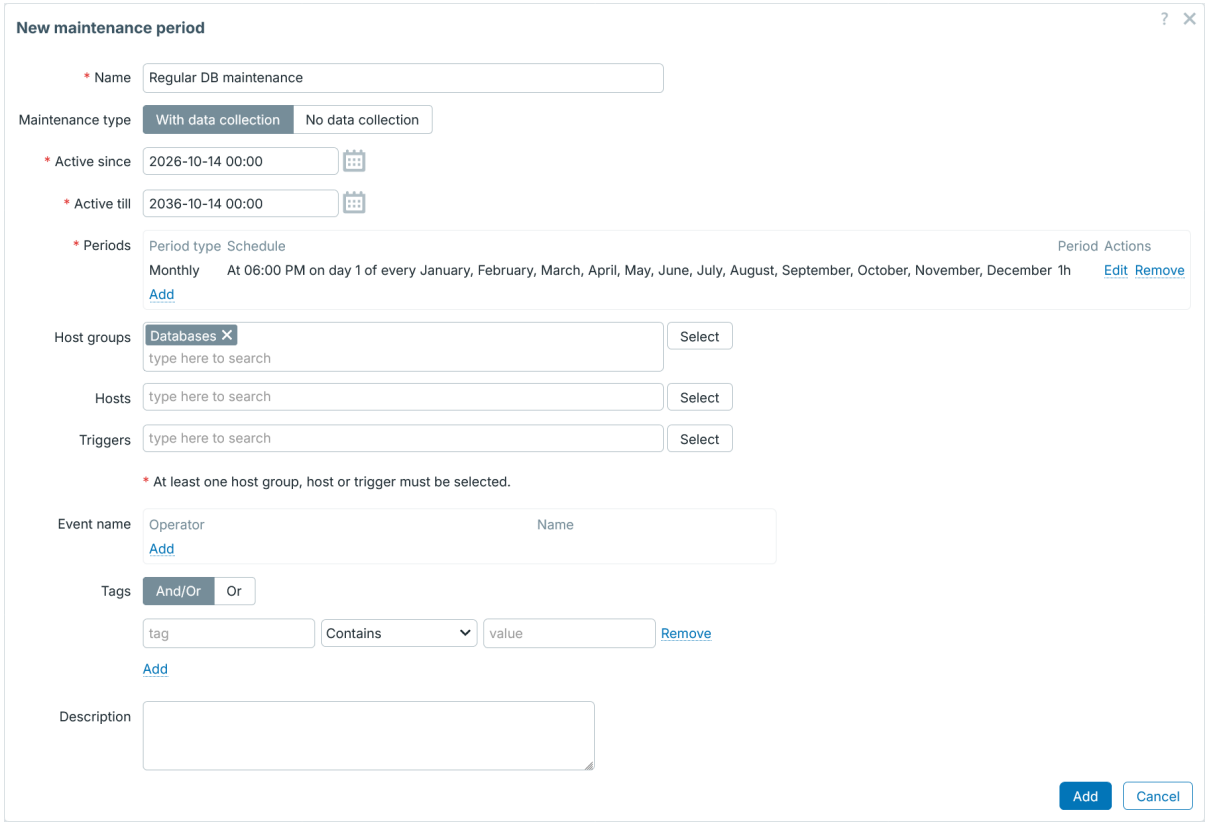
Настройка

Чтобы немедленно начать период обслуживания:

1. Перейдите в *Мониторинг* > *Проблемы*.
2. Нажмите имя проблемы.
3. Выберите вариант обслуживания в меню события (узел сети, триггер, имя события, теги).
4. Измените параметры обслуживания в форме.

Чтобы настроить запланированный период обслуживания:

1. Перейдите в *Сбор данных* > *Обслуживание*.
2. Нажмите *Создать период обслуживания* (или имя существующего периода обслуживания).
3. Введите параметры обслуживания в форме.



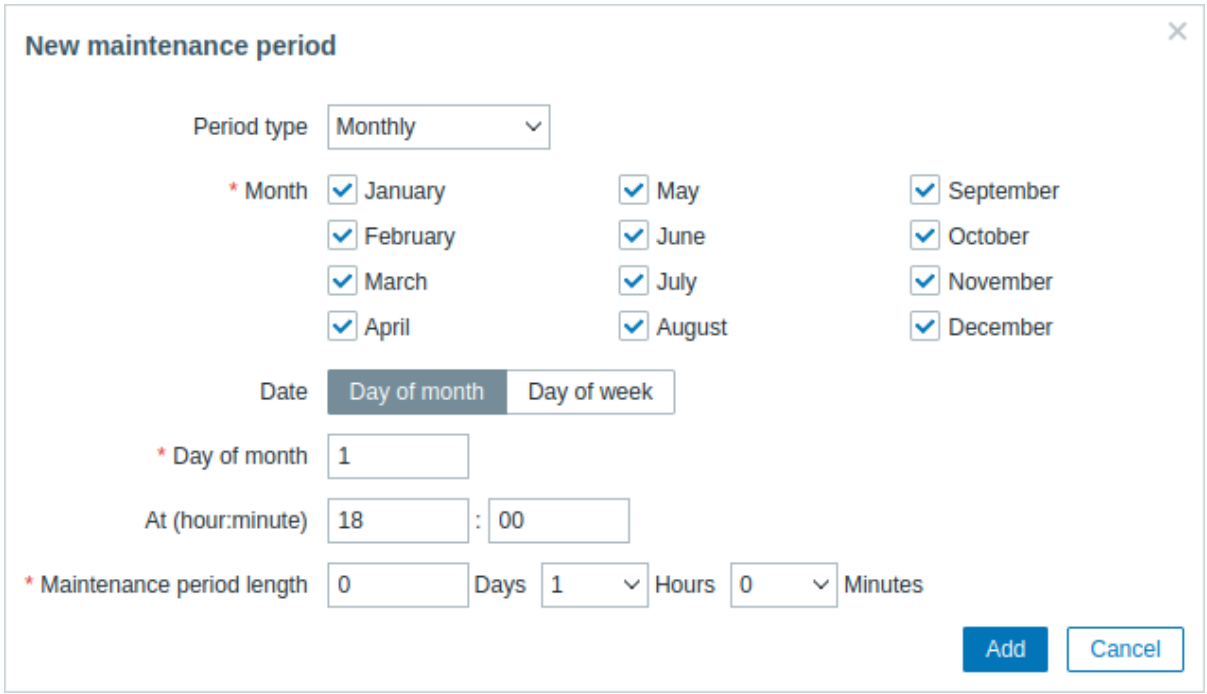
Все обязательные поля отмечены красной звездочкой.

Параметр	Описание
Имя	Имя периода обслуживания.
Тип обслуживания	Можно установить два типа обслуживания: **Со сбором данных** - во время обслуживания сервер собирает данные, а триггеры обрабатываются; **Без сбора данных** - данные могут по-прежнему собираться, но во время обслуживания они не сохраняются в базе данных, а триггеры (включая функцию `nodata()`) не срабатывают. Информацию о влиянии каждого типа на отчет о доступности см. в разделе Влияние периодов обслуживания.
Активен с	Дата и время, начиная с которых выполнение периодов обслуживания становится активным. *Примечание:* Установка только этого времени не активирует период обслуживания; периоды обслуживания необходимо настроить в разделе *Периоды* (см. ниже).
Активен до	Дата и время, до которых выполнение периодов обслуживания остается активным.
Периоды	Этот блок позволяет определить точные дни и часы проведения обслуживания. При нажатии Add открывается всплывающее окно с гибкой формой *Период обслуживания*, в которой можно задать расписание обслуживания. Подробное описание см. в разделе Периоды обслуживания.
Группы узлов сети	Выберите группы узлов сети, для которых будет активировано обслуживание. Обслуживание будет активировано для всех узлов сети из указанных групп узлов сети. Это поле поддерживает автодополнение: после начала ввода отображается раскрывающийся список всех доступных групп узлов сети. Указание родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети. Поэтому обслуживание также будет активировано для узлов сети из вложенных групп.
Узлы сети	Выберите узлы сети, для которых будет активировано обслуживание. Это поле поддерживает автодополнение: после начала ввода отображается раскрывающийся список всех доступных узлов сети.
Триггеры	Выберите триггеры, для которых будет активировано обслуживание. Это поле поддерживает автодополнение: после начала ввода отображается раскрывающийся список всех доступных триггеров.

Параметр	Описание
Имя события	Укажите имена событий, для которых будет активировано обслуживание. Можно задать несколько условий. Для каждого условия доступны два оператора: **Содержит** - включить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки без учета регистра) **Не содержит** - исключить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки без учета регистра)
Теги	Укажите теги, для которых будет активировано обслуживание. Можно задать несколько условий. Сопоставление имен тегов всегда выполняется с учетом регистра. Для каждого условия доступны несколько операторов: **Содержит** - включить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки с учетом регистра); **Равно** - включить указанные имена тегов и значения (с учетом регистра). **Не равно** - исключить указанные имена тегов и значения (с учетом регистра) **Не содержит** - исключить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки без учета регистра) Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега группируются с помощью условия ИЛИ; **ИЛИ** - достаточно выполнения одного условия. Теги можно указывать только при выборе типа обслуживания *Со сбором данных*.
Описание	Описание периода обслуживания.

Периоды обслуживания

Окно периода обслуживания предназначено для планирования времени для регулярного или разового обслуживания. Форма динамическая: доступные поля изменяются в зависимости от выбранного *Тип периода*.



Тип периода	Описание
Однократно	Настройте разовый период обслуживания: *Дата* - дата и время периода обслуживания; *Длительность периода обслуживания* - как долго обслуживание будет активно.

Тип периода	Описание
Ежедневно	Настройте ежедневный период обслуживания: *Каждые N дней* - частота обслуживания (1 - *(по умолчанию)* каждый день, 2 - каждые два дня и т. д.); *В (час:минута)* - время суток, когда начинается обслуживание; *Длительность периода обслуживания* - как долго обслуживание будет активно. Когда параметр *Каждые N дней* больше ”1”, начальным днем считается день, на который приходится время *Активно с*. Примеры: - если *Активно с* установлено в ”2021-01-01 12:00”, *Каждые N дней* установлено в ”2”, а *В (час:минута)* - в ”23:00”, то первый период обслуживания начнется 1 января в 23:00, а второй - 3 января в 23:00; - если *Активно с* установлено в ”2021-01-01 12:00”, *Каждые N дней* установлено в ”2”, а *В (час:минута)* - в ”01:00”, то первый период обслуживания начнется 3 января в 01:00, а второй - 5 января в 01:00.
Еженедельно	Настройте еженедельный период обслуживания: *Каждые N недель* - частота обслуживания (1 - *(по умолчанию)* каждую неделю, 2 - каждые две недели и т. д.); *День недели* - в какой день должно проводиться обслуживание; *В (час:минута)* - время суток, когда начинается обслуживание; *Длительность периода обслуживания* - как долго обслуживание будет активно. Когда параметр *Каждые N недель* больше ”1”, начальной неделей считается неделя, на которую приходится время *Активно с*. Примеры см. в описании параметра *Ежедневно* выше.
Ежемесячно	Настройте ежемесячный период обслуживания: *Месяц* - выберите все месяцы, в течение которых выполняется регулярное обслуживание; *Дата:* ***День месяца*** - выберите этот вариант, если обслуживание должно проводиться в одну и ту же дату каждого месяца (например, каждый 1-й день месяца), затем выберите нужный день в появившемся поле *День месяца*; *Дата:* ***День недели*** - выберите этот вариант, если обслуживание должно проводиться только в определенные дни (например, каждый первый понедельник месяца), затем выберите (в раскрывающемся списке) нужную неделю месяца (первую, вторую, третью, четвертую или последнюю), а затем отметьте флажками день(дни) обслуживания; *В (час:минута)* - время суток, когда начинается обслуживание; *Длительность периода обслуживания* - как долго обслуживание будет активно.

Attention:
При создании периода обслуживания используется часовой пояс пользователя, который его создает. Однако при планировании повторяющихся периодов обслуживания (*Daily*, *Weekly*, *Monthly*) используется часовой пояс сервера Zabbix. Чтобы обеспечить предсказуемое поведение повторяющихся периодов обслуживания, необходимо использовать единый часовой пояс для всех компонентов Zabbix.

Когда закончите, нажмите *Add*, чтобы добавить период обслуживания в раздел *Periods*.

Обратите внимание, что изменения времени на летнее/зимнее время (DST) не влияют на длительность обслуживания. Например, предположим, что у нас настроено двухчасовое обслуживание, которое обычно начинается в 01:00 и заканчивается в 03:00:

- если после одного часа обслуживания (в 02:00) происходит переход на летнее/зимнее время и текущее время меняется с 02:00 на 03:00, обслуживание продолжится еще один час (до 04:00);
- если после двух часов обслуживания (в 03:00) происходит переход на летнее/зимнее время и текущее время меняется с 03:00 на 02:00, обслуживание завершится, потому что прошло два часа;
- если период обслуживания начинается в час, который пропускается из-за перехода на летнее/зимнее время, то обслуживание не начнется.

Если период обслуживания установлен на ”1 day” (фактическая длительность обслуживания составляет 24 часа, поскольку Zabbix вычисляет дни в часах), начинается в 00:00 и заканчивается в 00:00 следующего дня:

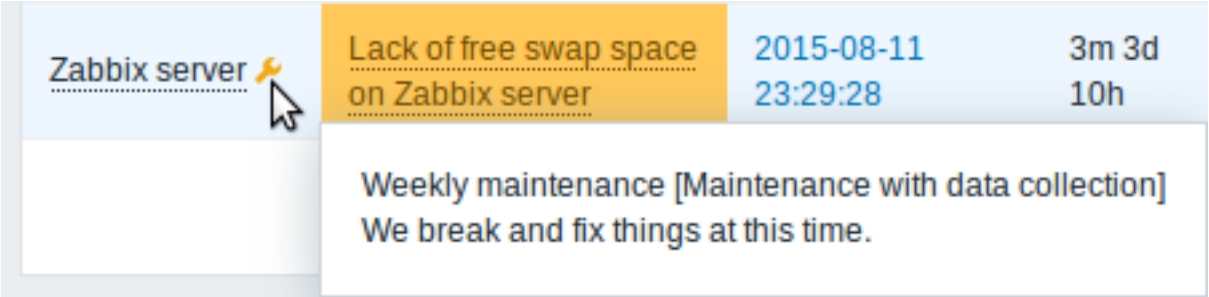
- обслуживание завершится в 01:00 следующего дня, если текущее время сдвинется вперед на один час;
- обслуживание завершится в 23:00 того же дня, если текущее время сдвинется назад на один час.

Отображение

Отображение узлов сети в обслуживании

Оранжевый значок гаечного ключа рядом с именем узла сети указывает на то, что этот узел сети находится в обслуживании в следующих разделах:

- *Панели*
- *Мониторинг* > *Проблемы*
- *Инвентаризация* > *Узлы сети* > *Детали инвентаризации узла сети*
- *Сбор данных* > *Узлы сети* (см. столбец *Статус*)



Сведения об обслуживании отображаются, когда указатель мыши находится над значком.

Кроме того, узлы сети в обслуживании выделяются оранжевым фоном в *Мониторинг* > *Карты*.

Отображение подавленных проблем

Обычно проблемы для узлов сети, находящихся на обслуживании, подавляются, то есть не отображаются в веб-интерфейсе.

Подавленные проблемы также могут быть скрыты, если они были подавлены вручную.

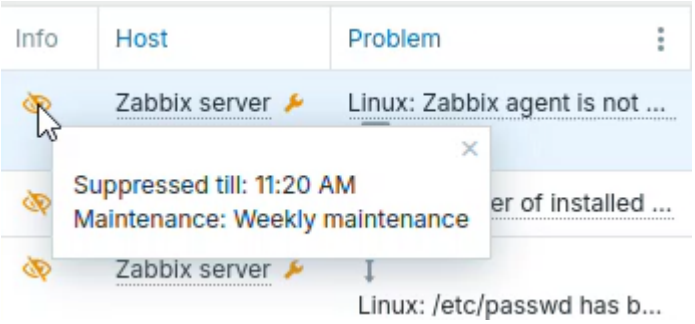
Отображать подавленные проблемы можно, выбрав параметр *Show suppressed problems* в следующих местах:

- *Dashboards* (в конфигурации виджета *Problem hosts*, *Problems*, *Problems by severity*, *Trigger overview*)
- *Monitoring* > *Problems* (в фильтре)
- *Monitoring* > *Maps* (в конфигурации карты)
- Глобальные notifications (в конфигурации профиля пользователя)

Когда отображаются подавленные проблемы, показывается следующий значок: . Если подавление связано с обслуживанием узла сети, отображаются следующие значки:

- - проблема была подавлена из-за обслуживания узла сети.
- - проблема была снята с подавления после обслуживания узла сети.

При наведении указателя на значок отображается дополнительная информация. В деталях события можно проверить, была ли событие подавлено из-за обслуживания или подавлено вручную. Если применяются и обслуживание, и ручное подавление, приоритет в цвете значка имеет ручное подавление.



Расчет очередей во время обслуживания

Очереди, отображаемые во веб-интерфейсе Zabbix (Administration > Queue), рассчитываются сервером Zabbix. Они не включают элементы данных в обслуживании *No data collection*, и длина очереди для этих элементов всегда равна нулю, даже если их значения задерживаются. Задержанные элементы данных в обслуживании *With data collection* по-прежнему учитываются в очереди.

Прокси Zabbix proxy не знает о периодах обслуживания, поскольку между сервером Zabbix и прокси отсутствует синхронизация конфигурации обслуживания. Внутренние проверки, вычисляемые на прокси Zabbix (например,

zabbix[queue,,] и zabbix[stats,,,queue,,]), сообщают о задержанных элементах данных независимо от статуса обслуживания на сервере Zabbix.

В результате для одних и тех же элементов данных в обслуживании *No data collection* веб-интерфейс Zabbix и внутренние проверки на прокси Zabbix могут сообщать разную длину очереди.

Тегирование

Теги служат нескольким целям, прежде всего для пометки событий. Когда сущности помечены тегами, любое новое событие, связанное с помеченной сущностью, будет наследовать ее теги. Например:

- для помеченных шаблонов - любая проблема узла сети (созданная триггерами из шаблона) будет наследовать теги шаблона.
- для помеченных узлов сети - любая проблема узла сети будет наследовать теги узла сети.
- для помеченных элементов данных/веб-сценариев - любая проблема элемента данных/веб-сценария будет наследовать теги элемента данных/веб-сценария.
- для помеченных триггеров - любая проблема, созданная триггером, будет наследовать теги триггера.

Событие проблемы наследует все теги из всей цепочки сущностей - шаблонов, узлов сети, элементов данных/веб-сценариев, триггеров. Идентичные комбинации tag:value (после подстановки макросов) объединяются в одну, что позволяет избежать дублирования.

События восстановления, сгенерированные при ручном закрытии, также включают разрешенные теги события, унаследованные от шаблонов, узлов сети, элементов данных/веб-сценариев и триггеров. Эти теги доступны в уведомлениях и через макросы, такие как {EVENT.RECOVERY.TAGS} и {EVENT.RECOVERY.TAGSJSON}.

Пользовательские теги событий обеспечивают большую гибкость. Например:

- корреляция событий может быть настроена на основе тегов событий.
- условия действий могут быть настроены на основе тегов событий.
- проблемы элементов данных могут группироваться на основе тегов событий.
- теги проблем могут использоваться для сопоставления проблем с сервисами.

Сущности могут быть помечены одним и тем же именем тега, но с разными значениями тега (например, component:memory и component:storage). Аналогично, у сущности может быть тег без значения и тот же тег со значением (например, database и database:postgresql). Такие теги не считаются дубликатами.

Теги состоят из имени тега и значения тега. При назначении тегов сущностям можно использовать только имя или пару имя-значение (например, mysql, jira, target:mysql, service:jira и т. д.).

Теги могут быть определены для различных сущностей:

- Шаблоны
- Узлы сети
- Элементы данных
- Веб-сценарии
- Триггеры
- Сервисы
- Элементы данных и триггеры шаблонов
- Прототипы узлов сети, элементов данных и триггеров

В списках *Шаблоны*, *Узлы сети*, *Элементы данных*, *Триггеры*, *Веб-сценарии* и их прототипах столбец *Теги* отображает как обычные, так и унаследованные теги. При наведении курсора на унаследованный тег или щелчке по нему появляется всплывающая подсказка JavaScript с текстом "Inherited tag". Если унаследованный тег также существует как обычный тег, отображается другой текст подсказки в зависимости от списка, где он показан (например, "Inherited and template tag" в списке шаблонов или "Inherited and host tag" в списке узлов сети). В других местах веб-интерфейса обычные и унаследованные теги отображаются вместе (дубликаты удаляются), без значков и без дополнительного текста подсказки.

Note:
Обратитесь к официальным рекомендациям Zabbix по общим рекомендациям по определению тегов, а также к отдельным рекомендациям для шаблонов, элементов данных, триггеров и правил обнаружения низкого уровня.

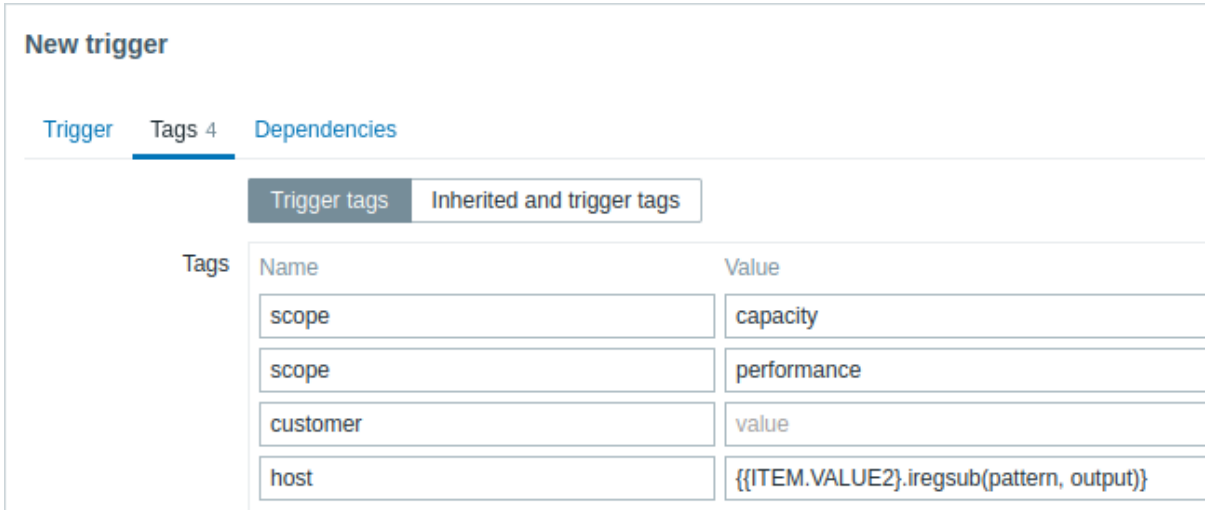
Сценарии использования Некоторые распространенные сценарии использования тегирования:

1. Пометка событий триггера:
   - Определите тег триггера (например, scope:performance).

- Проблемы, созданные этим триггером, будут иметь тег триггера.

2. Пометка проблем, унаследованных от шаблона:
   - Определите тег шаблона (например, `target:mysql`).
   - Проблемы, созданные триггерами из этого шаблона, будут иметь тег шаблона.
3. Пометка проблем узла сети:
   - Определите тег узла сети (например, `service:jira`).
   - Проблемы, созданные триггерами этого узла сети, будут иметь тег узла сети.
4. Фильтрация связанных элементов данных:
   - Определите тег элемента данных (например, `component:cpu`).
   - В *Monitoring* > *Latest data* элементы данных можно фильтровать по тегу `component:cpu`.
5. Использование информации, извлеченной из значения элемента данных, в качестве значения тега:
   - Определите тег с макросом в качестве значения тега (например, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - В *Monitoring* > *Problems* у проблем значение тега будет преобразовано в данные, извлеченные из значения элемента данных.
6. Определение проблем в файле журнала и их отдельное закрытие:
   - Определите тег триггера для триггера элемента данных мониторинга журнала, который будет извлекать значения из значения элемента данных с помощью макроса (например, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - В конфигурации триггера настройте корреляцию событий:
     - установите *PROBLEM event generation mode* в значение "Multiple";
     - установите *OK event closes* в значение "All problems if tag values match";
     - задайте тег для сопоставления.
   - Проблемы, созданные триггером элемента данных журнала, будут иметь тег триггера и будут закрываться по отдельности.
7. Фильтрация уведомлений:
   - Определите теги триггера (например, `scope:security` для trigger1 и `scope:availability` для trigger2).
   - Используйте фильтрацию по тегам в условиях действия, чтобы получать уведомления только по событиям, соответствующим данным тегов.
8. Определение проблем в уведомлениях:
   - Определите теги триггера.
   - Используйте макрос {EVENT.TAGS} в уведомлении о проблеме.
   - Уведомление о проблеме будет содержать теги триггера, что позволит легче определить, к какому приложению/службе относится уведомление.
9. Упрощение задач настройки с помощью тегов шаблона:
   - Определите тег триггера шаблона.
   - Триггеры, созданные из этого триггера шаблона, будут иметь его тег.
10. Создание триггеров с тегами из низкоуровневого обнаружения (LLD):
    - Определите тег прототипа триггера с макросом LLD в имени или значении тега (например, `scope:{#FSNAME}`).
    - Триггеры, созданные из прототипа триггера, будут иметь его тег.
11. Сопоставление сервисов с использованием тегов сервиса:
    - Определите теги сервиса.
    - Настройте действия сервиса для сервисов с совпадающими тегами.
    - Кроме того, используйте теги сервиса, чтобы связать сервис с SLA для расчетов SLA.
12. Связывание сервисов с проблемами с помощью тегов проблемы сервиса:
    - Определите тег проблемы в конфигурации сервиса (например, `target:mysql`).
    - Проблемы с совпадающим тегом будут автоматически сопоставлены с сервисом, а статус сервиса будет изменяться на основе настроенных правил расчета статуса сервиса.
13. Подавление проблем, когда узел сети находится в режиме обслуживания:
    - Определите теги в конфигурации периода обслуживания.
    - Проблемы с заданными тегами будут подавляться.
14. Предоставление доступа группам пользователей:
    - Определите теги в конфигурации группы пользователей.
    - Пользователи в группе пользователей смогут просматривать только проблемы с заданными тегами.

Конфигурация Теги можно определить на отдельной вкладке, например, в конфигурации триггера:



Поддержка макросов Встроенные и пользовательские макросы в тегах разрешаются в момент возникновения события. До того как событие произойдет, эти макросы будут отображаться в веб-интерфейсе Zabbix как неразрешенные.

Макросы низкоуровневого обнаружения разрешаются в процессе обнаружения.

Следующие макросы могут использоваться в именах и значениях тегов триггера:

- {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID} встроенные макросы
- {INVENTORY.*} встроенные макросы (для ссылки на значения инвентаризации узла сети с одного или нескольких узлов сети в выражении триггера)
- Пользовательские макросы и пользовательские макросы с контекстом (контекст может включать макросы низкоуровневого обнаружения)
- Макросы низкоуровневого обнаружения (только в тегах прототипа триггера)

Следующие макросы могут использоваться в именах и значениях тегов шаблона, узла сети и элемента данных/веб-сценария:

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID} встроенные макросы
- {INVENTORY.*} встроенные макросы
- Пользовательские макросы
- Макросы низкоуровневого обнаружения (только в тегах прототипа узла сети и элемента данных)

Следующие макросы могут использоваться в уведомлениях на основе триггеров:

- {EVENT.TAGS} и {EVENT.RECOVERY.TAGS} встроенные макросы (эти макросы будут преобразованы в список тегов события или тегов события восстановления, разделенный запятыми)
- {EVENT.TAGSJSON} и {EVENT.RECOVERY.TAGSJSON} встроенные макросы (эти макросы будут преобразованы в массив JSON, содержащий объекты тега события objects или объекты тега события восстановления)

Извлечение подстроки в тегах триггеров

Извлечение подстроки поддерживается для заполнения имени тега или значения тега с использованием функции макроса. Функция применяет регулярное выражение к значению, полученному из поддерживаемого макроса. Например:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

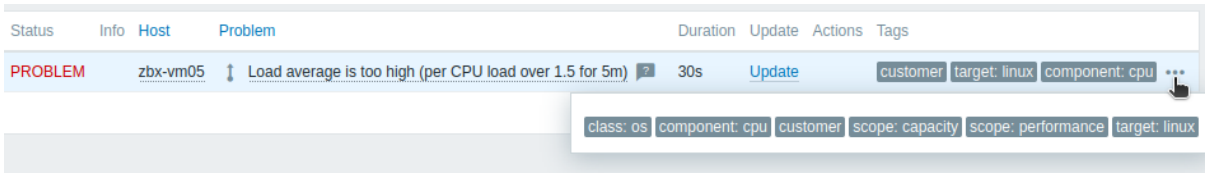
Если имя тега или значение после подстановки макросов превышает 255 символов, оно будет обрезано до 255 символов.

См. также: Использование функций макросов в макросах обнаружения на низком уровне для тегирования событий.

Просмотр тегов событий Теги, если они определены, можно просматривать вместе с новыми событиями в:

- *Мониторинг* > *Проблемы*
- *Мониторинг* > *Проблемы* > *Детали события*
- *Панели* > *Виджет "Проблемы"*

Порядок и количество отображаемых тегов определяются параметрами фильтрации *Приоритет отображения тегов* и *Показывать теги* в *Мониторинг* > *Проблемы* или в виджете *Проблемы* на панели. Обратите внимание, что может отображаться не более трех тегов; если тегов больше, при наведении на три точки отображаются все теги во всплывающем окне.



Complex event processing

Overview Complex event processing (CEP) performs rule-based processing for Zabbix events.

For example, you might want to:

- group related events and mark first event in a series as cause (the following as symptoms)
- specify a time-to-live period for events
- specify a maximum number for certain events
- be notified about similar events by location
- be notified if some event happens too often (or not at all) within a time period

In addition to single event processing, you may define **time windows** for event analysis. Within a time window, all matching events form a single context and can be analyzed as a whole.

A sliding time window moves forward with the time. Older events leave the window (are ”evicted”), while new events may enter.

A tumbling time window does not move forward. When the window expires, a new window with the same duration is started. *Cause and symptom grouping* windows are a tumbling time window.

The time window concept presents logical points for executing operations. For example, an event can be transformed or ”enriched” when it enters the window - severity can be raised or lowered, tags can be added or adjusted, suppression can be activated, etc.

Similar operations can be executed when the event leaves the window or when a pattern is found.

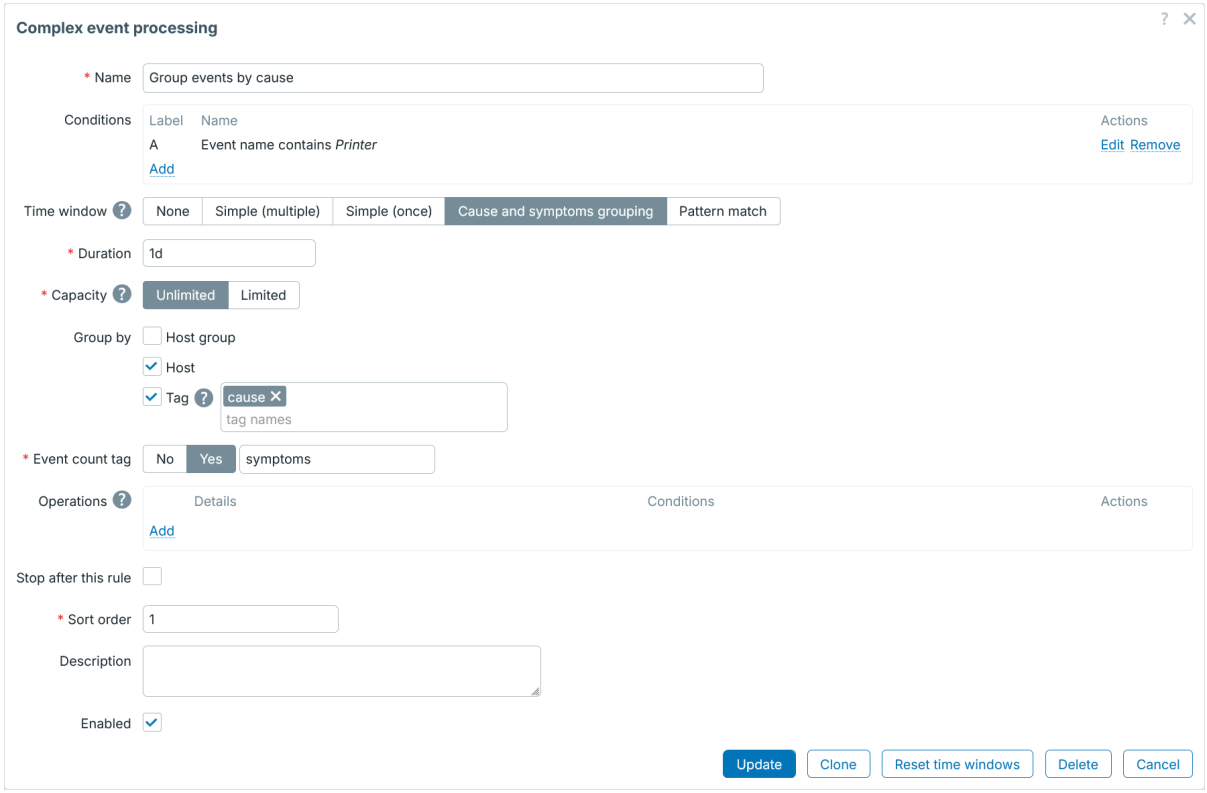
Event stream analysis is defined by CEP rules. A CEP rule consists of a time window (optional), conditions and operations. Operations are tied to an execution point.

Use cases

- Group related events, with the first such event automatically marked as cause (all following as symptoms).
- Enrich event with higher severity/specific tag if too many 401 events happen within a minute (possible bot activity).

Configuration To create a CEP rule in Zabbix frontend, do the following:

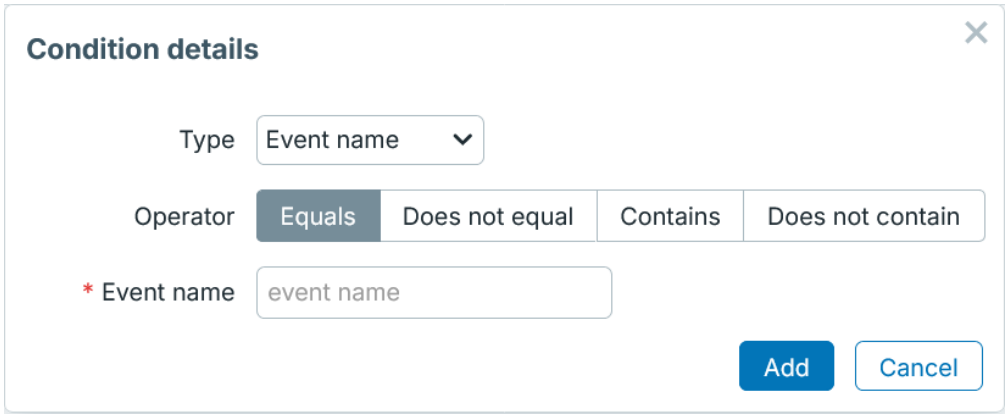
1. Go to *Data collection* > *Event processing*.
2. Click *Create complex event processing* in the upper-right corner of the screen.
3. Enter parameters of the rule in the form.



Parameter	Description
Name	Specify a unique CEP rule name.
Conditions	Specify conditions for associating events with this CEP rule.
Time window	Select the time window type: **None** - process single events with no time window (default) **Simple (multiple)** - process events in a sliding time window (event can be added to multiple windows of this type) **Simple (once)** - process events in a sliding time window (event can be added only to one window of this type) **Cause and symptom grouping** - process first event in time window as cause, consecutive events as symptoms **Pattern match** - analyze sliding time window events based on a JavaScript script. The window is examined every second; events older than the duration are evicted, and the script is then run over the events that remain in it. When the script reports a match, the pattern match operations are executed for every event the window holds; the events stay in the window and are examined again a second later, until they age out. A window can be forcibly closed by the ”Close window” operation.
Duration	Limit the time window duration (in seconds, 365 days max). The window duration is started by the first matching event. In *Simple* and *Pattern match* windows, the window is moving so older events will leave the window (are evicted) after the duration passes. In *Cause and symptom grouping* windows, window duration is a fixed timeframe for setting cause and symptom relationship. The window is started when the first event matches. When the duration expires, the window is closed and a new one is started. No event is evicted when the duration expires; periods run back to back. Time suffixes are supported (`10m`, `1h`, etc). User macros are supported.
Capacity	Specify the event capacity within the time window: **Unlimited** - event count is not limited **Limited** - event count is limited by the specified number Once the limit is surpassed, the latest event is evicted. User macros are supported.
Group by	Group events by host group, host, and/or event tag. This option is available for CEP rules with a time window. If specified, the rule will create multiple windows based on specified criteria; otherwise events will be grouped into a single window group.

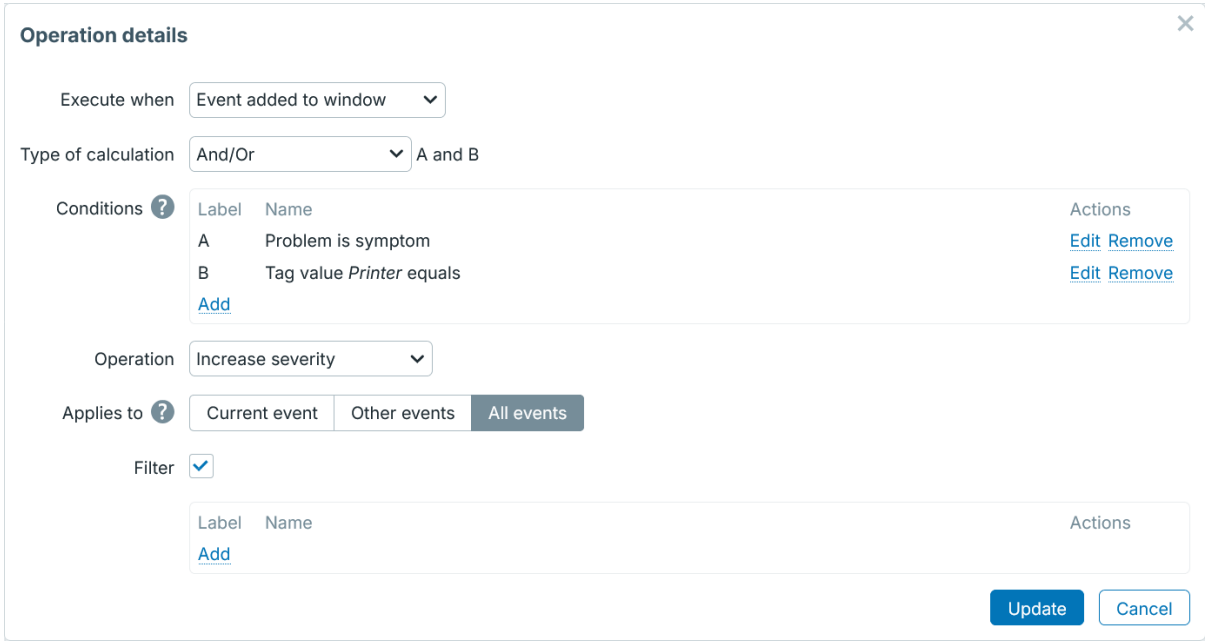
Parameter	Description
Event count tag	Specify name for a tag displaying symptom event count. Symptom event count will be displayed as `<tag name>:<event count>` (for example: `symptoms:2`). This field is only available if *Cause and symptom grouping* is selected as time window type.
Script	Specify the script for identifying event patterns. This field is only available if *Pattern match* is selected as time window type.
Operations	Click on *Add* to specify operations.
Stop after this rule	Mark the checkbox if no further processing is required after this rule. Otherwise, continue with the next CEP rule according to the sort order.
Sort order	Specify position of this CEP rule in the processing order. If the order priority is the same, the rules will be executed alphabetically.
Description	Add the CEP rule description.

CEP rule condition



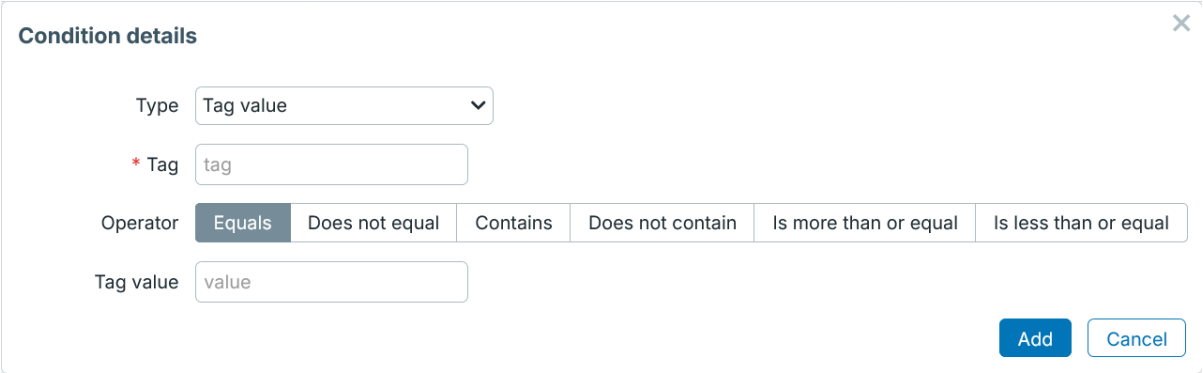
Parameter	Description
Type	Select condition type. Event filtering can be based on event name, tag name, tag value, severity, host group, host (visible name), or time period.

CEP rule operation



Parameter	Description
Execute when	Select execution point: **Event occurred** - when event occurs. This is the only stage where *Discard* operation is supported, since event is not actually created yet and can be discarded. Window-related operations are not available at this point. Changes made to the event in *Event occurred* phase (except closing) are visible to the next CEP rules trying to match this event. For example, if rule A added tag X and rule B has condition ”Tag X exists”, then it will match rule A and B. Of course, rule A should not be configured to stop after matching. **Event added to window** - when event is added to time window. Operations are executed after event is added to matched CEP-rule windows. If window is at its capacity, then *Event evicted* operations are executed instead. **Event evicted** - when event leaves the window. Operations are executed before event is evicted from a window because of duration or capacity. **Event closed** - when event is closed. **Window closed** - when time window is closed. Operations are executed on all window events when window is closed because of *Close window* operation or as expired cause-symptom window. **Pattern matched** - when pattern is matched. Operations are executed on all window events when the pattern match JavaScript returns `true`. Note that the *Event occurred*, *Event added to window* and *Event evicted* have fixed execution order, while *Window closed* and *Pattern matched* operate independently.
Conditions	Select additional conditions for the operation.
Operation	Select operation. **Events**: *Set name* - set event name *Close* - close event *Discard* - discard event. Only available at *Event occurred* stage. *Set severity* - set event severity *Increase severity* - increase event severity *Decrease severity* - decrease event severity *Suppress* - suppress event *Unsuppress* - unsuppress event *Clone first* - clone first event. Useful for producing a summary event based on the first event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. *Clone last* - clone last event. Useful for producing a summary event based on the last event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. For event cloning to work either the source (first/last) event must be resolved or its trigger must have multi-event generation enabled. **Tags**: *Add* - add tag *Set* - set tag and value *Set value* - set tag value *Increase value* - increase tag value *Decrease value* - decrease tag value *Rename* - rename tag *Remove* - remove tag **Window**: *Close (window)* - close event window
Applies to	Select operation scope: **Current event** - apply operation to the current event **Other events** - apply operation to all events except the current one **All events** - apply operation to all events
Filter	Select additional filter for the operation scope. This field is available if *Applies to* is set ”Other events” or ”All events”.

CEP rule operation condition



Parameter	Description
Type	Select condition type. Event filtering can be based on tag name, tag value, problem is opened/closed, problem is symptom/not symptom, first event in time window, last event in time window, problem is suppressed/not suppressed, event is cloned/not cloned.

Корреляция событий

Обзор

Корреляция событий позволяет сопоставлять события проблем с их устранением очень точно и гибко.

Корреляцию событий можно определить:

- На уровне триггера - один триггер можно использовать для сопоставления отдельных проблем с их решением;
- Глобально - проблемы можно сопоставлять с их решением, полученным от другого триггера или метода опроса, с помощью глобальных правил корреляции.

Корреляция событий на основе триггеров

Обзор

Корреляция событий на основе триггеров позволяет сопоставлять отдельные проблемы, о которых сообщает один триггер.

Обычно событие OK может закрыть все события проблем, созданные одним триггером, однако в некоторых случаях требуется более детальный подход. Например, при мониторинге файлов журналов может потребоваться обнаруживать определенные проблемы в файле журнала и закрывать их по отдельности, а не все сразу.

Это относится к триггерам, у которых параметр *PROBLEM event generation mode* установлен в значение *Multiple*. Такие триггеры обычно используются для мониторинга журналов, обработки ловушек и т. д.

В Zabbix можно связывать события проблем на основе тегирования. Теги используются для извлечения значений и создания идентификаторов событий проблем. Благодаря этому проблемы также можно закрывать по отдельности на основе совпадения тегов.

Иными словами, один и тот же триггер может создавать отдельные события, идентифицируемые тегом события. Поэтому события проблем можно идентифицировать по одному и закрывать отдельно на основе идентификации по тегу события.

Как это работает

При мониторинге журналов вы можете столкнуться со строками, похожими на следующие:

```
Line1: Service 1 stopped
Line2: Service 2 stopped
Line3: Service 1 was restarted
Line4: Service 2 was restarted
```

Идея корреляции событий заключается в том, чтобы сопоставить событие проблемы из Line1 с решением из Line3, а событие проблемы из Line2 с решением из Line4, и закрывать эти проблемы по одной:

```
Line1: Service 1 stopped
Line3: Service 1 was restarted #problem from Line 1 closed

Line2: Service 2 stopped
Line4: Service 2 was restarted #problem from Line 2 closed
```

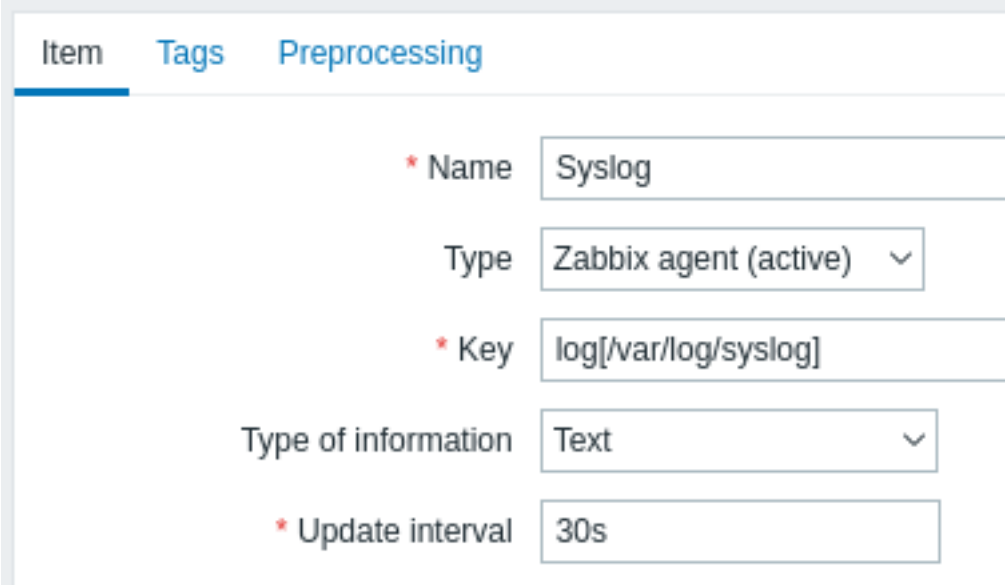
Для этого нужно пометить эти связанные события, например, как "Service 1" и "Service 2". Это можно сделать, применив регулярное выражение к строке журнала, чтобы извлечь значение тега. Затем, когда события создаются, им присваиваются теги "Service 1" и "Service 2" соответственно, и проблема может быть сопоставлена с решением.

Конфигурация

Элемент данных

Для начала вы можете настроить элемент данных, который отслеживает файл журнала, например:

```
log[/var/log/syslog]
```



После настройки элемента данных подождите минуту, чтобы изменения конфигурации были применены, а затем перейдите в Последние данные, чтобы убедиться, что элемент данных начал собирать данные.

Триггер

Когда элемент данных работает, вам нужно настроить триггер. Важно определить, какие записи в файле журнала заслуживают внимания. Например, следующее выражение триггера будет искать строку вроде 'Stopping', чтобы сигнализировать о потенциальных проблемах:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Чтобы каждая строка, содержащая строку "Stopping", считалась проблемой, также установите для *Режима генерации событий Problem* в настройках триггера значение 'Multiple'.

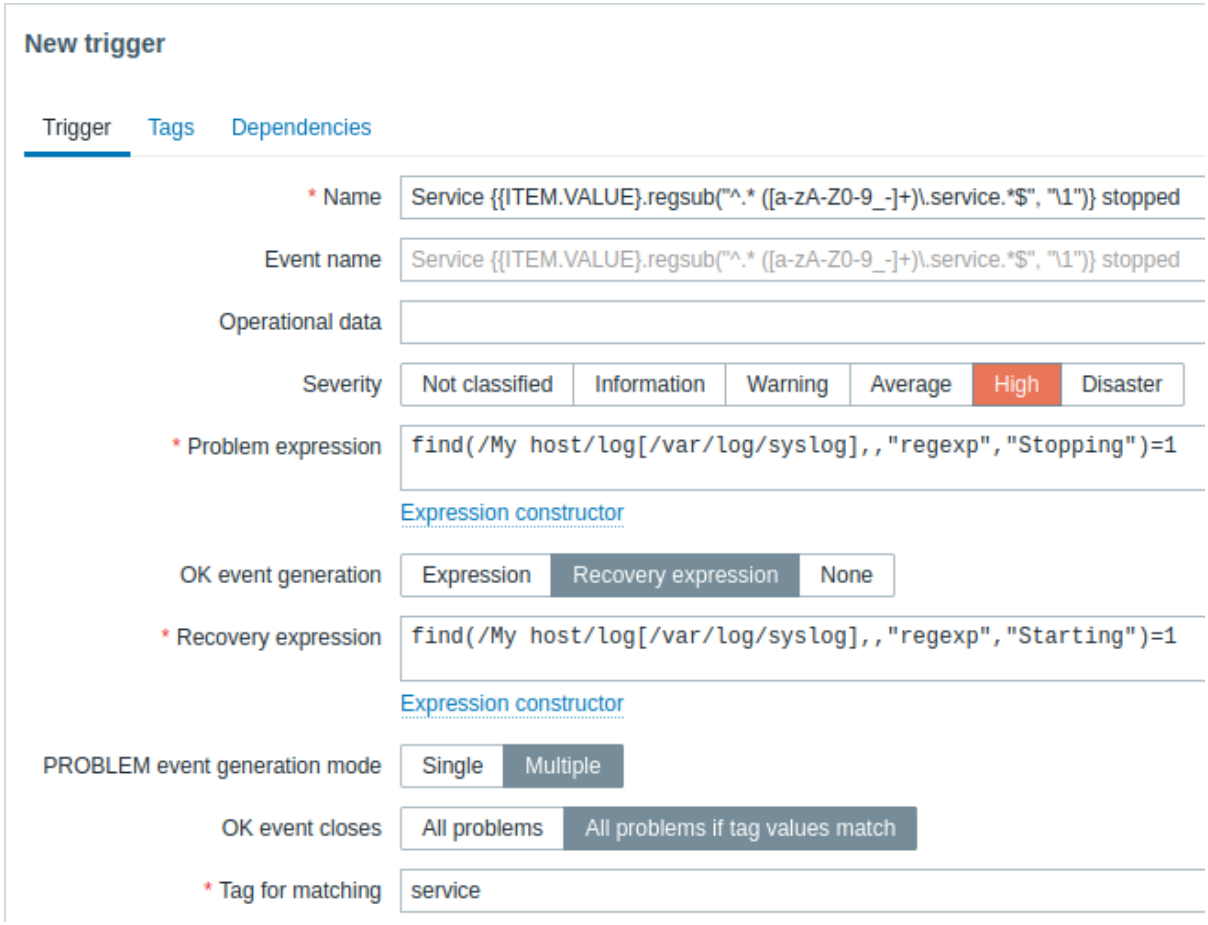
Затем задайте выражение восстановления. Следующее выражение восстановления устранит все проблемы, если в строке журнала будет найдена строка "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

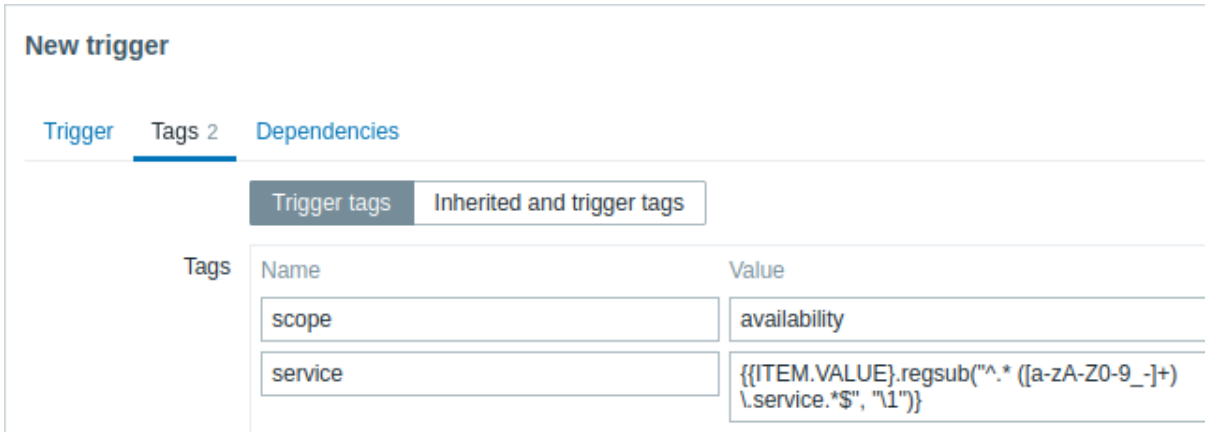
Поскольку мы этого не хотим, важно каким-либо образом убедиться, что соответствующие корневые проблемы закрываются, а не только все проблемы. Здесь может помочь тегирование.

Проблемы и восстановления можно сопоставлять, указав тег в настройках триггера. Необходимо задать следующие параметры:

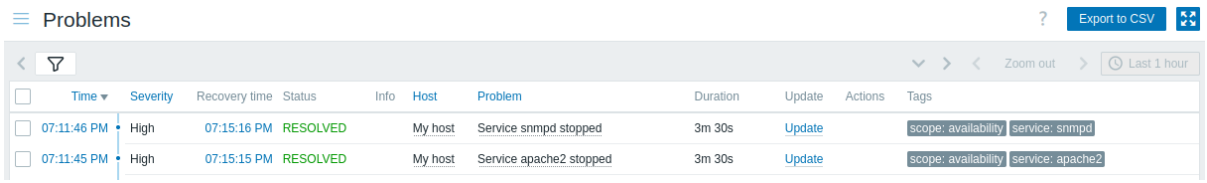
- *Режим генерации событий Problem*: Multiple
- *Событие OK закрывает*: Все проблемы, если значения тегов совпадают
- Введите имя тега для сопоставления событий



- настройте теги, чтобы извлекать значения тегов из строк журнала



Если настройка выполнена успешно, вы сможете видеть события проблем, помеченные по приложению и сопоставленные с их устранением в *Monitoring* > *Problems*.



Warning:
Поскольку возможна неверная настройка, при которой похожие теги событий могут создаваться для **несвязанных** проблем, пожалуйста, ознакомьтесь с приведенными ниже случаями!

- Индексированные макросы всегда относятся к полю *Expression* в настройках триггера, а не к *Recovery expression*. Например, в событии восстановления **{ITEM.VALUE1}** будет разрешен в последнее значение первого элемента данных **в выражении проблемы** на момент восстановления. Если выражение восстановления основано на другом элементе данных, а значение элемента данных из выражения проблемы изменится к моменту восстановления,

события получат разные теги и не будут сопоставлены.

- Если два приложения записывают сообщения об ошибках и восстановлении в один и тот же файл журнала, пользователь может решить использовать два тега *service* в одном триггере с разными значениями тегов, применяя отдельные регулярные выражения в значениях тегов для извлечения имен, скажем, service A и service B из макроса {ITEM.VALUE} (например, когда форматы сообщений различаются). Однако это может не сработать так, как планировалось, если регулярные выражения не совпадут. Несовпавшие регулярные выражения приведут к пустым значениям тегов, а одного пустого значения тега и в событиях проблемы, и в событиях OK достаточно для их сопоставления. Поэтому сообщение о восстановлении от service A может случайно закрыть сообщение об ошибке от service B.
- Фактические теги и значения тегов становятся видимыми только при срабатывании триггера. Если используемое регулярное выражение недопустимо, оно без предупреждения заменяется строкой *UNKNOWN*. Если исходное событие проблемы со значением тега *UNKNOWN* было пропущено, могут появиться последующие события OK с тем же значением тега *UNKNOWN*, которые закроют события проблемы, которые не должны были быть закрыты.
- Если пользователь использует макрос {ITEM.VALUE} без функций макросов в качестве значения тега, применяется ограничение в 255 символов. Когда сообщения журнала длинные, а первые 255 символов не содержат специфики, это также может привести к похожим тегам событий для несвязанных проблем.

Глобальная корреляция событий

Обзор

Глобальная корреляция событий позволяет охватить все метрики, отслеживаемые Zabbix, и создавать корреляции.

Можно сопоставлять события, созданные совершенно разными триггерами, и применять к ним одинаковые операции. Создавая интеллектуальные правила корреляции, можно избавиться от тысяч повторяющихся уведомлений и сосредоточиться на первопричинах проблемы!

Глобальная корреляция событий — это мощный механизм, который позволяет отказаться от логики проблем и восстановления, основанной на одном триггере. До сих пор одно событие проблемы создавалось одним триггером, и для устранения проблемы требовался тот же триггер. Нельзя было устранить проблему, созданную одним триггером, с помощью другого триггера. Но благодаря корреляции событий на основе тегов это стало возможным.

Например, триггер журнала может сообщать о проблемах приложения, а триггер опроса — о том, что приложение работает. Используя теги событий, можно присвоить триггеру журнала тег *status:down*, а триггеру опроса — тег *status:up*. Затем в глобальном правиле корреляции можно связать эти триггеры и назначить для этой корреляции соответствующую операцию, например закрытие старых событий.

В другом случае глобальная корреляция может выявлять похожие триггеры и применять к ним одну и ту же операцию. Что, если получать только один отчет о проблеме для каждой проблемы сетевого порта? Нет необходимости сообщать о них всех. Глобальная корреляция событий также позволяет это сделать.

Глобальная корреляция событий настраивается в **правилах корреляции**. Правило корреляции определяет, как новые события проблем сопоставляются с существующими событиями проблем и что делать при совпадении (закрыть новое событие или закрыть совпавшие старые события, создав соответствующие события OK). Если проблема закрыта глобальной корреляцией, это отображается в столбце *Информация* раздела *Мониторинг* > *Проблемы*.

Настраивать правила глобальной корреляции могут только пользователи уровня «Суперадминистратор».

Attention:
Корреляцию событий необходимо настраивать очень внимательно, поскольку она может отрицательно повлиять на производительность обработки событий или, при неправильной настройке, закрыть больше событий, чем предполагалось (в худшем случае могут быть закрыты все события проблем).

Чтобы **безопасно** настроить глобальные правила корреляции, соблюдайте следующие важные рекомендации:

- Уменьшайте область корреляции. Всегда задавайте уникальный тег для нового события, сопоставляемого со старыми событиями, и используйте условие корреляции *Имя тега нового события*.
- Добавляйте явное условие для старого события при использовании **Закрыть старые события**. Всегда добавляйте хотя бы одно условие *Старое событие* (например, *Имя тега старого события*, *Значение тега старого события* или *Пара тегов события*) при выборе **Закрыть старые события**; в противном случае правило может сопоставить и закрыть несвязанные существующие события проблем (в худшем случае — все проблемы). Для сопоставления динамических значений (host:port, идентификатор сеанса и т. д.) предпочтительно использовать *Пару тегов события*, а при возможности дополнительно ограничивать сопоставление узлом сети или группой узлов сети.

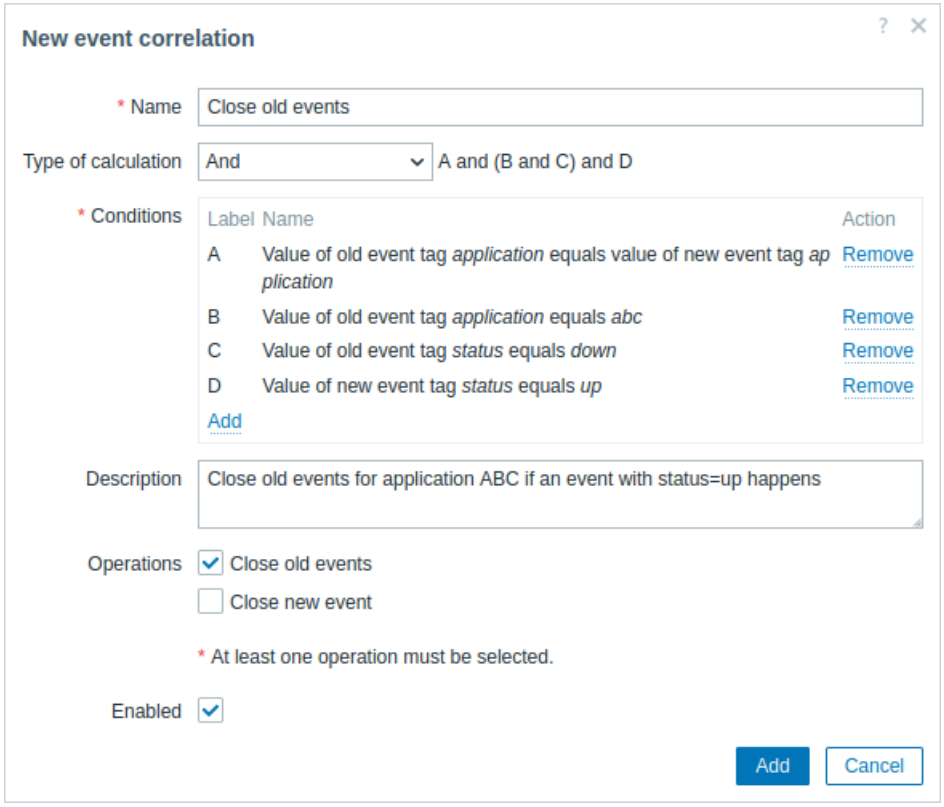
- Избегайте использования общих имен тегов, которые могут в итоге применяться в разных конфигурациях корреляции.
- Ограничивайте количество правил корреляции только действительно необходимыми.

См. также: известные проблемы.

Конфигурация

Чтобы глобально настроить правила корреляции событий:

1. Перейдите в *Сбор данных* > *Корреляция событий*.
2. Нажмите *Создать корреляцию событий* справа (или на имя корреляции, чтобы изменить существующее правило).
3. Введите параметры правила корреляции в форме.

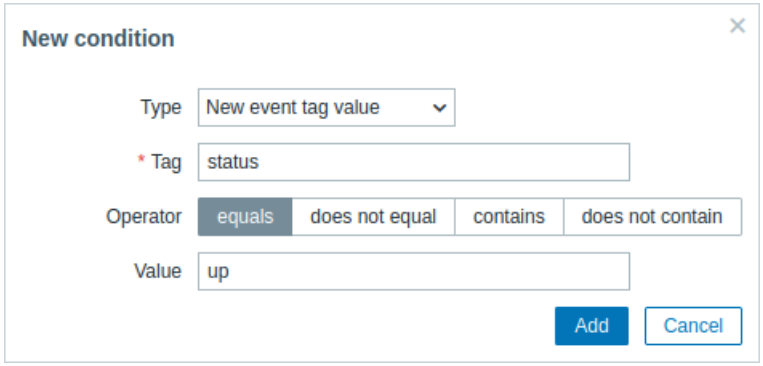


Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя	Уникальное имя правила корреляции.
Тип вычисления	Доступны следующие варианты вычисления условий: **And** - должны быть выполнены все условия **Or** - достаточно выполнения одного условия **And/Or** - AND для разных типов условий и OR для одинакового типа условий **Custom expression** - пользовательская формула вычисления для оценки условий действия. Она должна включать все условия (представленные заглавными буквами A, B, C, ...) и может содержать пробелы, табуляции, скобки ( ), **and** (с учетом регистра), **or** (с учетом регистра), **not** (с учетом регистра).
Conditions	Список условий. Подробности по настройке условия см. ниже.
Description	Описание правила корреляции.

Параметр	Описание
Operations	Отметьте флажок операции, которую нужно выполнить, когда событие скоррелировано. Доступны следующие операции: **Close old events** - закрыть старые события при возникновении нового события. Всегда добавляйте условие на основе старого события при использовании операции *Close old events*, иначе могут быть закрыты все существующие проблемы. **Close new event** - закрыть новое событие при его возникновении. **Warning!** Не оставляйте условия старого/нового события пустыми при использовании *Close old events*/*Close new event*. Если вы выберете операцию *Close old events* без добавления условия, соответствующего старому событию, Zabbix может сопоставить все существующие старые события и закрыть их. Всегда добавляйте явное условие старого события (например, *Old event tag name* или *Event tag pair*) при использовании *Close old events*. Например, правило, которое использует только *New event condition* и операцию *Close old events*, будет сопоставлять все старые события, которые соответствуют (отсутствующим) критериям старого события, - фактически закрывая старые проблемы.
Enabled	Если вы отметите этот флажок, правило корреляции будет включено.

Чтобы настроить подробности нового условия, нажмите Add в блоке Conditions. Откроется всплывающее окно, где можно изменить параметры условия.



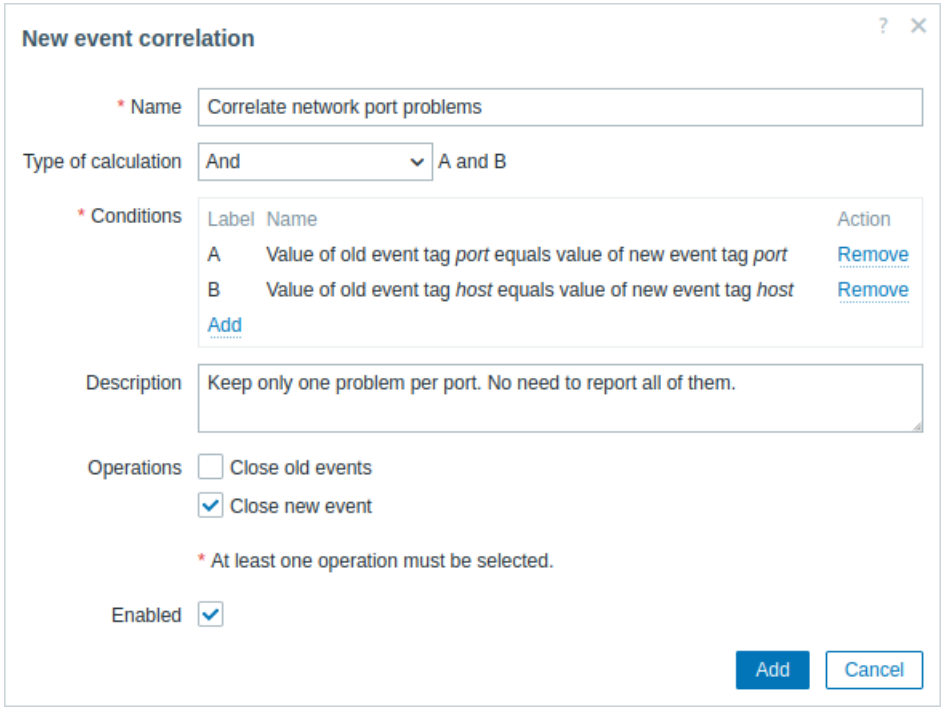
Параметр	Описание
New condition	Выберите условие для корреляции событий. *Обратите внимание*, что если не указано условие старого события, могут быть сопоставлены и закрыты все старые события. Аналогично, если не указано условие нового события, могут быть сопоставлены и закрыты все новые события. Доступны следующие условия: **Old event tag name** - укажите имя тега старого события для сопоставления. **New event tag name** - укажите имя тега нового события для сопоставления. **New event host group** - укажите группу узлов сети нового события для сопоставления. **Event tag pair** - укажите имя тега нового события и имя тега старого события для сопоставления. В этом случае совпадение будет, если совпадают **значения** тегов в обоих событиях. **Имена** тегов могут не совпадать. Этот вариант полезен для сопоставления значений во время выполнения, которые могут быть неизвестны на момент настройки (см. также Example). **Old event tag value** - укажите имя тега старого события и значение для сопоставления, используя следующие операторы: *equals* - имеет значение тега старого события *does not equal* - не имеет значения тега старого события *contains* - содержит строку в значении тега старого события *does not contain* - не содержит строку в значении тега старого события **New event tag value** - укажите имя тега нового события и значение для сопоставления, используя следующие операторы: *equals* - имеет значение тега нового события *does not equal* - не имеет значения тега нового события *contains* - содержит строку в значении тега нового события *does not contain* - не содержит строку в значении тега нового события

Warning:
Поскольку возможна неправильная настройка, а похожие теги событий могут создаваться для **не связанных** проблем, пожалуйста, ознакомьтесь с приведенными ниже случаями!

- Фактические теги и значения тегов становятся видимыми только при срабатывании триггера. Если используемое регулярное выражение некорректно, оно без предупреждения заменяется строкой *UNKNOWN*. Если начальное проблемное событие со значением тега *UNKNOWN* пропущено, могут появиться последующие события OK с тем же значением тега *UNKNOWN*, которые могут закрыть проблемные события, которые не должны были быть закрыты.
- Если пользователь использует макрос {ITEM.VALUE} без функций макросов в качестве значения тега, применяется ограничение в 255 символов. Когда сообщения журнала длинные и первые 255 символов не содержат специфики, это также может привести к появлению похожих тегов событий для не связанных проблем.

Пример

Останавливайте повторяющиеся события проблем с одного и того же сетевого порта.



Это глобальное правило корреляции будет сопоставлять проблемы, если значения тегов *host* и *port* присутствуют у триггера и совпадают в исходном событии и новом.

Операция закроет новые события проблем на том же сетевом порту, оставив открытым только исходную проблему.

Обнаружение

Обзор Функция обнаружения предоставляет автоматические способы добавления узлов сети и настройки мониторинга без ручного конфигурирования. Zabbix поддерживает несколько методов обнаружения, что позволяет адаптироваться к различным сценариям.

Методы обнаружения

- Обнаружение сети - автоматически обнаруживать узлы сети и службы, доступные в вашей сети.
- Автоматическая регистрация активного агента - разрешить активным агентам автоматически регистрироваться на сервере.
- Низкоуровневое обнаружение - автоматически создавать элементы данных, триггеры, графики и узлы сети для обнаруженных сущностей, таких как файловые системы или сетевые интерфейсы, с использованием прототипов и вложенных правил обнаружения.

Обнаружение сети

Обзор

Zabbix предлагает функцию автоматического обнаружения сети, которая отличается эффективностью и высокой гибкостью.

При правильной настройке обнаружения сети вы можете:

- Ускорить развертывание Zabbix.
- Упростить администрирование.
- Использовать Zabbix в быстро меняющихся средах без чрезмерных затрат на администрирование.

Обнаружение сети в Zabbix основано на следующей информации:

- Диапазоны IP-адресов
- Доступность внешних служб (FTP, SSH, WEB, POP3, IMAP, TCP и т. д.)
- Информация, полученная от агента Zabbix (поддерживается только незашифрованный режим)
- Информация, полученная от SNMP-агента

Оно НЕ предоставляет обнаружение топологии сети.

Обнаружение сети в основном состоит из двух фаз: обнаружения и действий.

Обнаружение

Zabbix периодически сканирует диапазоны IP-адресов, определенные в правилах сетевого обнаружения. Частота проверки настраивается отдельно для каждого правила.

Для каждого правила определяется набор проверок служб, которые должны выполняться для диапазона IP-адресов.

Правила обнаружения обрабатываются менеджером обнаружения. Менеджер обнаружения создает задание для каждого правила со списком задач (сетевых проверок). Сетевые проверки выполняются параллельно доступными рабочими процессами обнаружения (их количество настраивается во веб-интерфейсе отдельно для каждого правила). Исключение составляют проверки SNMPv3, которые обрабатываются одним рабочим процессом. Последовательно планируются только проверки с одинаковыми IP-адресом и портом, поскольку некоторые устройства не принимают параллельные подключения к одному и тому же порту.

Размер очереди сетевых проверок ограничен примерно 2000000 или 4 ГБ памяти. Если очередь переполнится, правило обнаружения будет пропущено, а в журнале будет выведено предупреждение. Для мониторинга количества проверок обнаружения в очереди можно использовать внутренний элемент данных `zabbix[discovery_queue]`.

Проверки обнаружения обрабатываются независимо от других проверок. Если какие-либо проверки не находят службу (или завершаются с ошибкой), остальные проверки все равно будут выполнены.

Note:
Если правило обнаружения изменяется во время выполнения, текущее выполнение обнаружения будет прервано.

Каждая проверка службы и узла сети (IP), выполняемая модулем сетевого обнаружения, генерирует событие обнаружения.

Событие	Результат проверки службы
Служба обнаружена	Служба находится в состоянии 'up' после состояния 'down' или обнаружена впервые.
Служба up	Служба находится в состоянии 'up', после того как уже была в состоянии 'up'.
Служба потеряна	Служба находится в состоянии 'down' после состояния 'up'.
Служба down	Служба находится в состоянии 'down', после того как уже была в состоянии 'down'.
Узел сети обнаружен	По крайней мере одна служба узла сети находится в состоянии 'up' после того, как все службы этого узла сети были в состоянии 'down', или обнаружена служба, относящаяся к незарегистрированному узлу сети.
Узел сети up	По крайней мере одна служба узла сети находится в состоянии 'up', после того как по крайней мере одна служба уже была в состоянии 'up'.
Узел сети потерян	Все службы узла сети находятся в состоянии 'down' после того, как хотя бы одна была в состоянии 'up'.
Узел сети down	Все службы узла сети находятся в состоянии 'down', после того как они уже были в состоянии 'down'.

Действия

События обнаружения могут быть основой соответствующих действий, таких как:

- Отправка оповещений
- Добавление/удаление узлов сети
- Активация/деактивация узлов сети
- Добавление узлов сети в группу
- Удаление узлов сети из группы
- Добавление тегов к узлу сети
- Удаление тегов с узла сети
- Присоединение шаблона к узлам сети/отсоединение шаблона от узлов сети
- Выполнение удалённых скриптов

Эти действия можно задать с учётом типа устройства, IP, состояния, времени доступности/недоступности и т.п. Для получения более подробных сведений о настройке действий на события, основанных на сетевом обнаружении, смотрите страницы операций действий и условий действий.

Поскольку действия на сетевое обнаружение основаны на событиях, они будут срабатывать и когда обнаруженный узел сети активен (online), и когда неактивен (offline). Крайне рекомендуется добавить условие действия *Состояние обнаружения: доступен*, чтобы избежать ситуации, когда такие действия как *Добавить узел сети* срабатывали бы при событиях *Сервис потерян* / *Сервис недоступен*. В противном случае, если обнаруженный узел сети удаляется вручную, по-прежнему будут генерироваться события *Сервис потерян* / *Сервис недоступен*, и он будет пересоздан во время следующего цикла обнаружения.

Note:
Присоединение обнаруженного узла сети к шаблонам будет массово заканчиваться неудачей, если любой из присоединяемых шаблонов имеет уникальные объекты (например, ключ элемента данных), совпадающие с уникальными объектами (т.е. ключом элемента данных), уже присутствующими на этом узле сети или в другом присоединяемом шаблоне.

Создание узла сети

Узел сети добавляется, если выбрана операция *Добавить узел сети*. Узел сети также добавляется, даже если операция *Добавить узел сети* отсутствует, если вы выбираете операции, приводящие к действиям над узлом сети. Такие операции:

- Включить узел сети.
- Отключить узел сети.
- Добавить узел сети в группу узлов сети.
- Связать шаблон с узлом сети.

Созданные узлы сети добавляются в группу *Discovered hosts* (по умолчанию, настраивается в *Administration* > *General* > *Other*). Если вы хотите, чтобы узлы сети добавлялись в другую группу, добавьте операцию *Remove from host groups* (указав ”Discovered hosts”), а также добавьте операцию *Add to host groups* (указав другую группу узлов сети), поскольку узел сети должен принадлежать группе узлов сети.

IP-адрес обнаруженного устройства вместе с источником обнаружения (сервер Zabbix, прокси Zabbix или группа прокси) и типом интерфейса используется как критерий для поиска узла сети в системе. Если узел сети с тем же IP-адресом, типом интерфейса и источником обнаружения уже существует, именно он будет целевым объектом для выполнения операций. Если источник обнаружения отличается, обнаруженный объект рассматривается как другой узел сети, и может быть создан новый узел сети.

Если IP-адрес обнаруженного узла сети изменится или интерфейс будет удален, при следующем обнаружении будет создан новый узел сети.

Именование узлов сети

При добавлении узлов сети имя узла сети является результатом обратного преобразования DNS или IP адресом, если обратное преобразование завершилось неудачей. Преобразование выполняется на стороне Zabbix сервера или Zabbix прокси, в зависимости от того, кто выполняет обнаружение. Если преобразование завершается неудачей на прокси, оно не повторяется на стороне сервера. Если узел сети с таким же именем уже существует, следующий узел сети получит добавление к имени **_2**, затем **_3** и так далее.

Также возможно переопределить преобразования DNS/IP и вместо этого для имени узла сети использовать значение элемента данных, например:

- Вы можете обнаружить несколько серверов с помощью Zabbix агента, используя элемент данных Zabbix агента для обнаружения, и автоматически назначить им правильное имя на основании строкового значения, возвращаемого этим элементом данных

- Вы можете обнаружить несколько сетевых устройств SNMP, используя для обнаружения элемент данных агента SNMP, и автоматически назначить им соответствующие имена на основании строкового значения, возвращаемого этим элементом данных

Если имя узла сети было назначено, используя значение элемента данных, оно не обновляется во время последующих проверок обнаружения. Если невозможно назначить имя узла сети, используя значение элемента данных, то используется значение по умолчанию (DNS имя).

Если узел сети с обнаруженным IP адресом уже существует и источник обнаружения (Zabbix сервер, прокси или группа прокси) не изменился, новый узел сети не создаётся. Если источник обнаружения отличается, обнаруженный объект считается другим и может быть создан новый узел сети. Однако, если действие на обнаружение содержит операции (присоединение шаблона, добавление в группу узлов сети и так далее), они будут выполнены над существующим узлом сети, который имеет совпадающие IP адрес, тип интерфейса и источник обнаружения.

Удаление узлов сети

Узлы сети, которые созданы правилом сетевого обнаружения, удаляются автоматически из *Мониторинг* → *Обнаружение* (*Monitoring* → *Discovery*), если обнаруженный объект более не входит в диапазон IP адресов этого правила. Узлы сети удаляются немедленно.

Создание интерфейсов при добавлении узлов сети

Когда узлы сети добавляются в результате сетевого обнаружения, для них создаются интерфейсы в соответствии со следующими правилами:

- Обнаруженные службы — например, если проверка SNMP завершилась успешно, будет создан SNMP-интерфейс.
- Если узел сети отвечал и на запросы Zabbix agent, и на запросы SNMP, будут созданы оба типа интерфейсов.
- Если критерии уникальности — это данные, возвращаемые Zabbix agent или SNMP, то первым найденным для узла сети интерфейсом будет создан интерфейс по умолчанию. Другие IP-адреса будут проигнорированы, если только интерфейс по умолчанию не выйдет из строя. Если интерфейс по умолчанию недоступен, в качестве замены для интерфейса по умолчанию будет использован наименьший из остальных IP-адресов. Условия действия (например, Host IP) не влияют на добавление интерфейсов. *Примечание*: это будет работать, если все интерфейсы обнаружены одним и тем же правилом обнаружения. Если другое правило обнаружения обнаружит другой интерфейс того же узла сети, будет добавлен дополнительный узел сети.
- Если узел сети отвечал только на проверки agent, он будет создан только с интерфейсом agent. Если позже он начнет отвечать на SNMP, будут добавлены дополнительные SNMP-интерфейсы.
- Если изначально были созданы 3 отдельных узла сети, обнаруженные по критерию уникальности ”IP”, а затем правило обнаружения было изменено так, что для узлов A, B и C результат критерия уникальности стал одинаковым, B и C будут созданы как дополнительные интерфейсы для A, первого узла. Отдельные узлы B и C при этом останутся. В *Monitoring* > *Discovery* добавленные интерфейсы будут отображаться в столбце ”Discovered device” черным шрифтом и с отступом, а в столбце ”Monitored host” будет отображаться только A, первый созданный узел. ”Uptime/Downtime” не измеряется для IP-адресов, которые считаются дополнительными интерфейсами.

Изменение настройки прокси

Узлы сети, обнаруженные разными прокси, не всегда считаются разными узлами сети. Проверки обнаружения и уникальности зависят от структуры группы прокси: когда прокси выполняет правило обнаружения и создает узел сети, этот узел сети добавляется в родительскую группу прокси, а не назначается самому прокси. Когда Zabbix проверяет уникальность IP-адресов во время обнаружения, он проверяет узлы сети, контролируемые родительской группой прокси. Узлы сети, контролируемые отдельными прокси в этой группе (включая прокси, который выполнил обнаружение), игнорируются при проверке уникальности, что может привести к появлению дублирующихся узлов сети, если несколько прокси контролируют пересекающиеся подсети.

Хотя такое поведение позволяет выполнять обнаружение в пересекающихся диапазонах IP-адресов, используемых разными подсетями, изменение прокси, назначенного уже контролируемой подсети, является более сложной задачей, поскольку изменения прокси должны последовательно применяться как к обнаруженным узлам сети, так и к составу родительской группы прокси, чтобы избежать дубликатов.

Например, шаги для замены прокси в правиле обнаружения:

1. Отключите правило обнаружения.
2. Синхронизируйте конфигурацию прокси.
3. Замените прокси в правиле обнаружения.
4. Замените прокси для всех узлов сети, обнаруженных этим правилом (убедитесь, что узлы сети в родительской группе прокси и любые узлы сети, контролируемые отдельными прокси в этой группе, обновлены, чтобы избежать дубликатов).
5. Включите правило обнаружения.

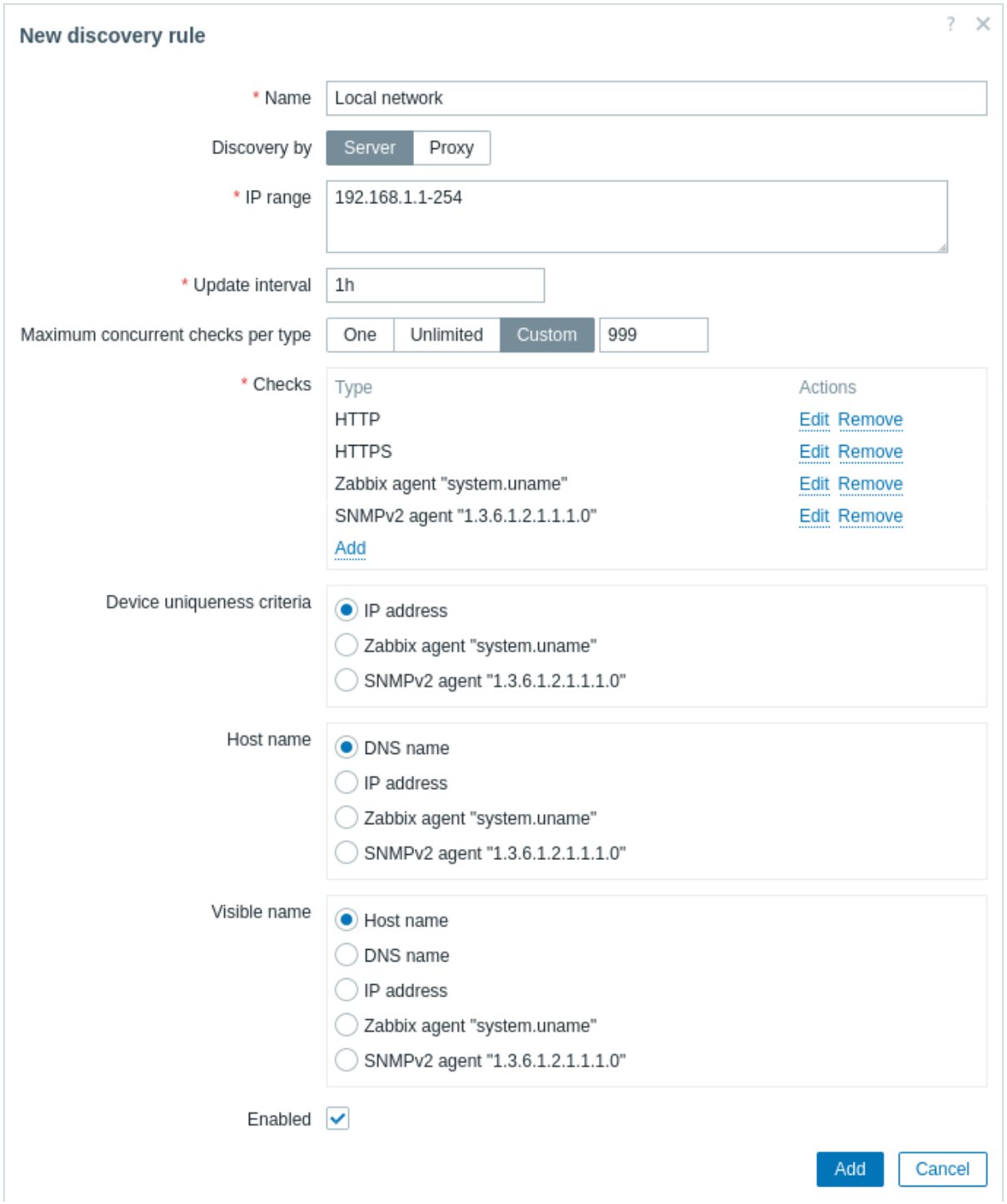
Настройка правила обнаружения сети

Обзор

Чтобы настроить правило обнаружения сети, используемое Zabbix для обнаружения узлов сети и служб:

1. Перейдите в *Сбор данных* > *Обнаружение*.
2. Нажмите *Создать правило обнаружения* (или имя правила, чтобы изменить существующее).
3. Отредактируйте атрибуты правила обнаружения.

Атрибуты правила



Все обязательные поля ввода отмечены красной звездочкой.

Parameter	Description
Name	Уникальное имя правила. Например, ”Local network”.

Parameter	Description
Discovery by	Обнаружение выполняется: **Server** - сервером Zabbix **Proxy** - прокси Zabbix (выбранным в поле имени прокси)
IP range	Диапазон IP-адресов для обнаружения. Он может иметь следующие форматы: Один IP-адрес: 192.168.1.33 Диапазон IP-адресов: 192.168.1-10.1-255. Диапазон ограничен общим количеством охватываемых адресов (менее 64K). Маска IP: 192.168.4.0/24 поддерживаемые маски IP: /16 - /30 для IPv4-адресов /112 - /128 для IPv6-адресов Список: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Это поле поддерживает пробелы, табуляцию и несколько строк.
Update interval	Этот параметр определяет, как часто Zabbix будет выполнять правило. Интервал отсчитывается после завершения выполнения предыдущего экземпляра обнаружения, поэтому наложения не происходит. Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. *Примечание*: если используется пользовательский макрос и его значение изменяется (например, 1w → 1h), следующий опрос будет выполнен в соответствии с предыдущим значением (в примере - в более отдаленном будущем).
Maximum concurrent checks per type	Установите максимальное число потоков обнаружения (workers) на проверку службы для параллельной обработки проверок обнаружения: **One** - один поток **Unlimited** - неограниченное число потоков (но не больше, чем в параметре StartDiscoverers) **Custom** - задайте пользовательское число потоков (0-999). Асинхронные проверки служб SNMPv1 и SNMPv2 обрабатываются одновременно доступными workers. Все правила обнаружения с асинхронными проверками служб SNMPv3 обрабатываются одним worker из-за особенностей реализации libsnmp; поэтому увеличение числа workers не приведет к повышению скорости обнаружения.
Checks	Zabbix будет использовать этот список проверок для обнаружения. Нажмите Add, чтобы настроить новую проверку во всплывающем окне. Поддерживаемые проверки: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping. Обнаружение на основе протокола использует функциональность **net.tcp.service[]** для проверки каждого узла сети, за исключением SNMP, который запрашивает SNMP OID. Агент Zabbix проверяется путем запроса элемента данных в незашифрованном режиме. Подробнее см. элементы данных агента. Параметр ’Ports’ может иметь один из следующих видов: Один порт: 22 Диапазон портов: 22-45 Список: 22-45,55,60-70 Начиная с Zabbix 7.0, все проверки служб выполняются асинхронно, за исключением проверок LDAP. Начиная с Zabbix 7.0, проверка HTTP/HTTPS выполняется через libcurl. Если Zabbix server/proxy собран без libcurl, проверки HTTP будут работать как в предыдущих версиях (то есть как проверки TCP), но проверки HTTPS работать не будут.
Device uniqueness criteria	Критерии уникальности могут быть следующими: **IP address** - не обрабатывать несколько устройств с одним IP-адресом. Если устройство с таким же IP уже существует, оно будет считаться уже обнаруженным, и новый узел сети не будет добавлен. **<discovery check>** - либо проверка агента Zabbix, либо проверка SNMP agent. Обратите внимание, что критерии уникальности, используемые при обнаружении, не совпадают с идентификацией узла сети в системе при выполнении действий. Критерии уникальности при обнаружении определяют, являются ли два или более обнаруженных устройства одинаковыми (или разными), тогда как в Zabbix критерием идентификации узла сети является только IP-адрес (см. Создание узла сети).

Parameter	Description
Host name	Задайте техническое имя узла сети создаваемого узла сети с использованием: **DNS name** - DNS-имя (по умолчанию) **IP address** - IP-адрес **<discovery check>** - полученное строковое значение проверки обнаружения (например, проверка агента Zabbix, SNMP agent check) См. также: Именование узла сети.
Visible name	Задайте видимое имя узла сети создаваемого узла сети с использованием: **Host name** - техническое имя узла сети (по умолчанию) **DNS name** - DNS-имя **IP address** - IP-адрес **<discovery check>** - полученное строковое значение проверки обнаружения (например, проверка агента Zabbix, SNMP agent check) См. также: Именование узла сети.
Enabled	Если флажок установлен, правило активно и будет выполняться сервером Zabbix. Если флажок снят, правило неактивно. Оно не будет выполняться.

Превышение лимита файловых дескрипторов

При большом количестве одновременных проверок возможно исчерпание лимита файловых дескрипторов для менеджера обнаружения.

Количество файловых дескрипторов, необходимых для обнаружения, равно числу рабочих процессов обнаружения * 1000. По умолчанию используется 5 рабочих процессов обнаружения, а мягкий лимит системы составляет примерно 1024.

Если этот лимит приближается, Zabbix уменьшит значение по умолчанию для числа одновременных проверок каждого типа для каждого рабочего процесса и запишет предупреждение в файл журнала. Однако если пользователь задал значение *Maximum concurrent checks per type* выше, чем значение, рассчитанное Zabbix, Zabbix будет использовать заданное пользователем значение для одного рабочего процесса.

Сценарий из реальной жизни

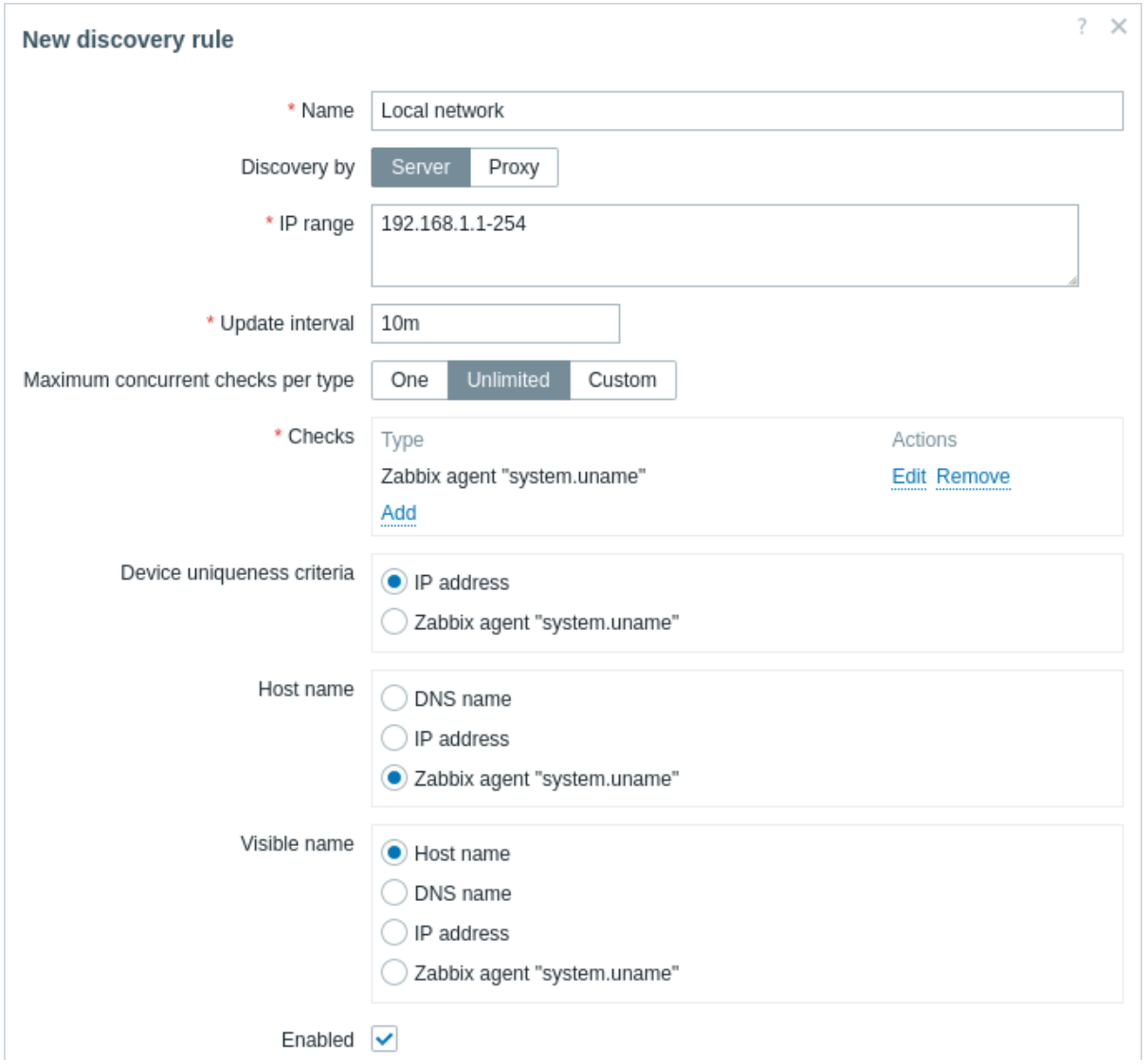
В этом примере мы бы хотели настроить сетевое обнаружение для локальной сети, имеющей диапазон IP адресов 192.168.1.1-192.168.1.254.

В нашем сценарии мы хотим:

- обнаруживать те узлы сети, на которых имеется Zabbix агент
- выполнять обнаружение каждые 10 минут
- добавлять узел сети для наблюдения, если время работы узла сети более 1 часа
- удалять узел сети, если он недоступен на протяжении более 24 часов
- добавлять узлы сети Linux в группу «Linux servers»
- добавлять узлы сети Windows в группу «Windows servers»
- использовать шаблон *Linux* для узлов сети Linux
- использовать шаблон *Windows* для узлов сети Windows

Шаг 1

Добавим правило сетевого обнаружения для нашего диапазона IP адресов.



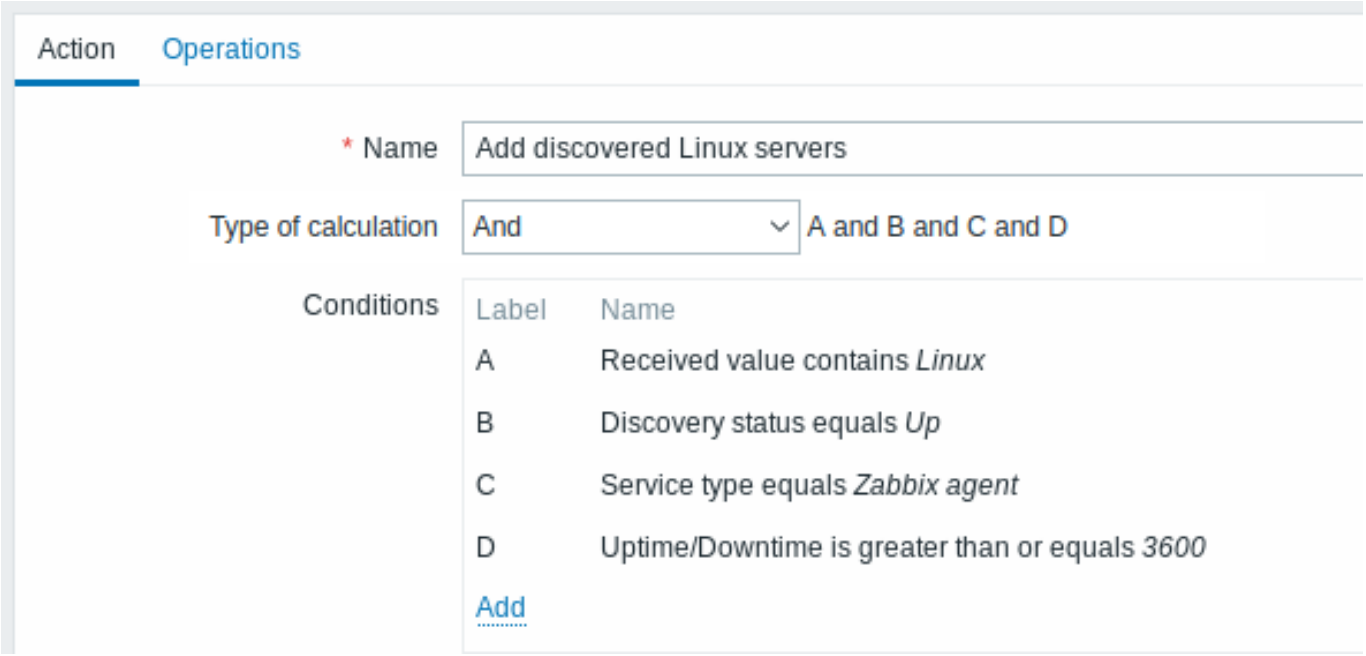
Zabbix будет пытаться обнаружить узлы сети в диапазоне IP адресов 192.168.1.1-192.168.1.254, подключаясь к Zabbix агенту и получая значение ключа **system.uname**. Полученное от агента значение может быть использовано для именования узла сети, а также для выполнения различных действий для разных операционных систем. Например, для присоединения шаблона *Windows* к серверам Windows, а шаблона *Linux* — к серверам Linux.

Правило будет выполняться каждые 10 минут.

Когда правило будет добавлено, Zabbix автоматически запустит обнаружение и порождение событий, основанных на обнаружении, для их дальнейшей обработки.

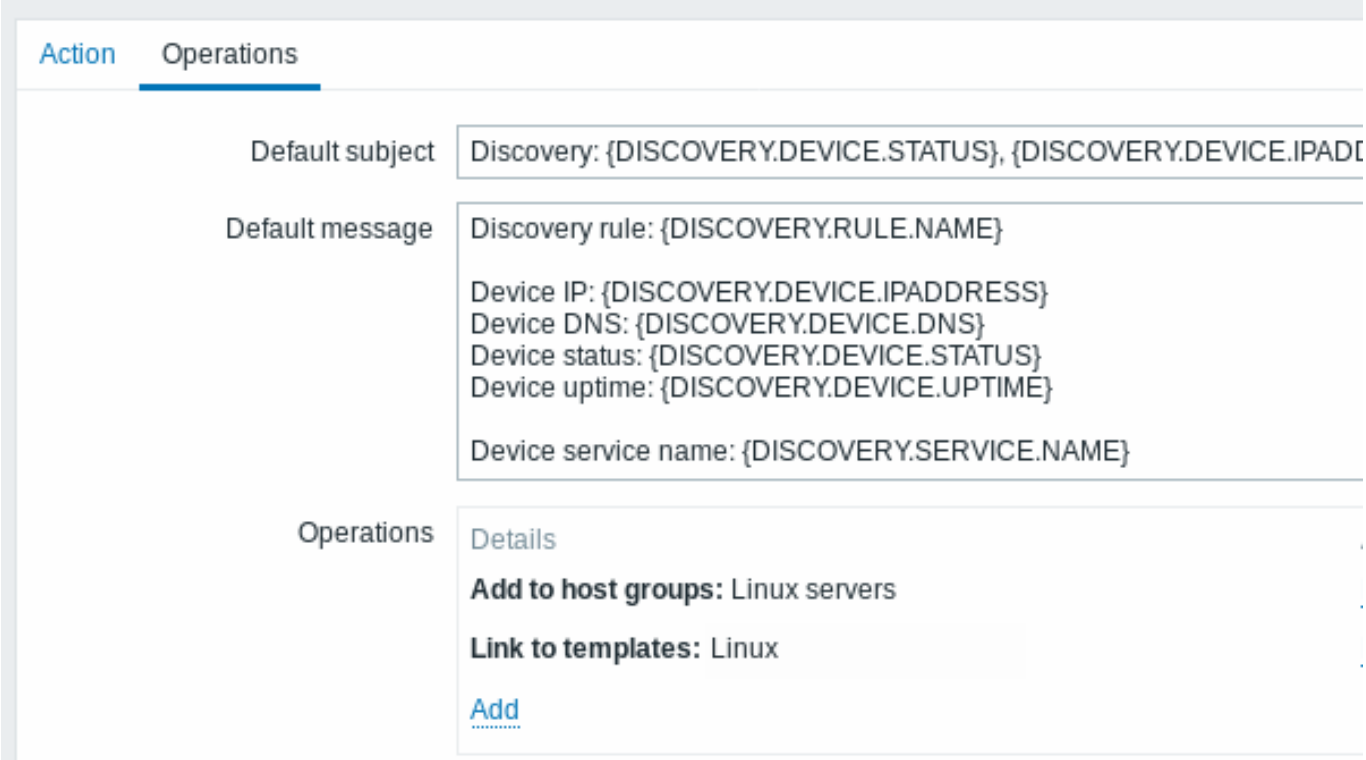
Шаг 2

Определим действие для добавления обнаруженных Linux серверов в соответвующие группы/шаблоны.



Это действие будет активировано, если:

- сервис «Zabbix agent» в состоянии «доступен»
- значение system.uname (ключ Zabbix агента, который мы использовали при создании правила) содержит «Linux»
- время работы — 1 час (3600 секунд) или больше

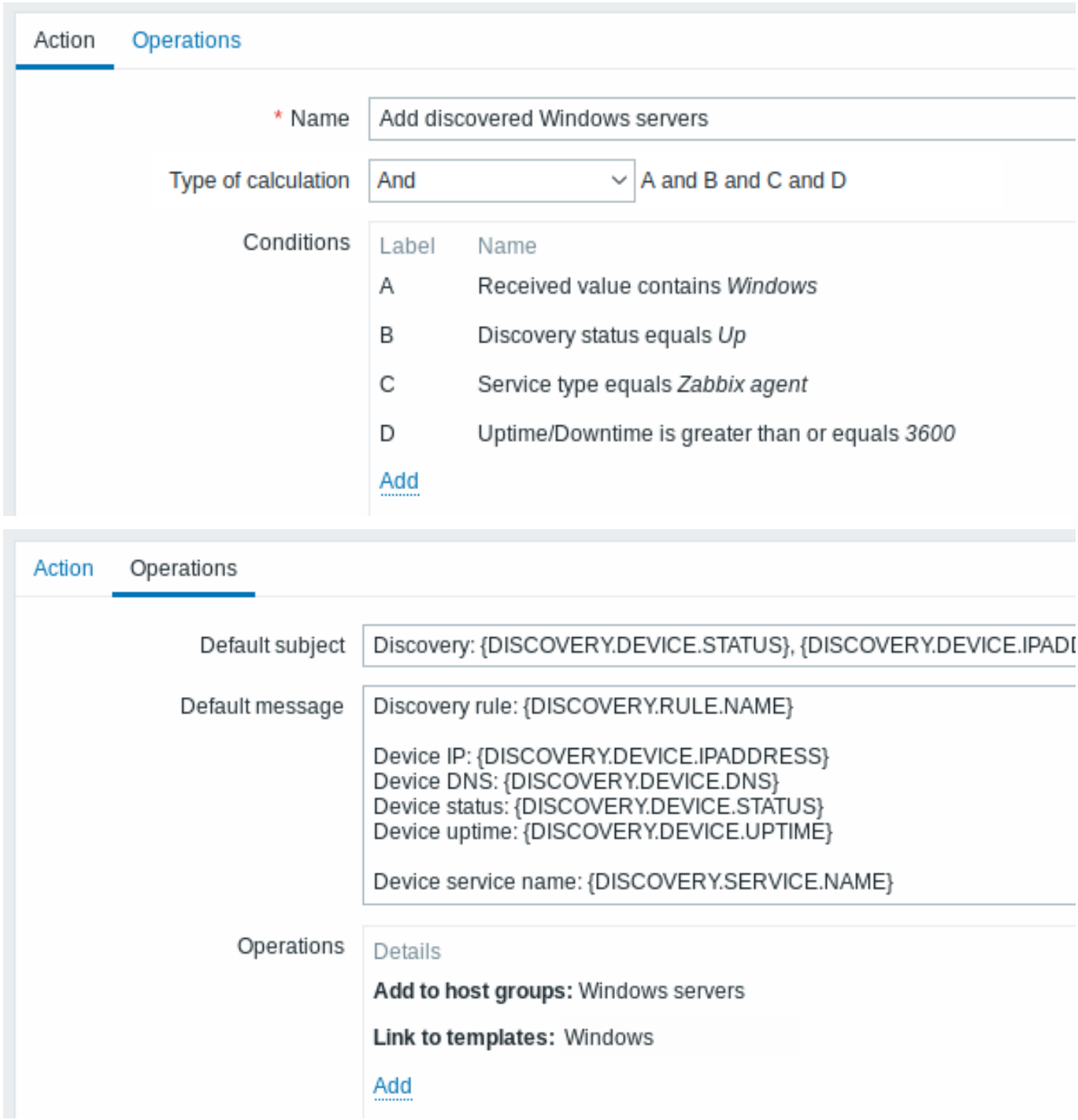


Это действие будет выполнять следующие операции:

- добавление обнаруженного узла сети в группу «Linux servers» (а также добавление узла сети, если он не был добавлен ранее)
- присоединение узла сети к шаблону *Linux*. Zabbix автоматически запустит наблюдение за узлом сети, используя элементы данных и триггеры из шаблона *Linux*.

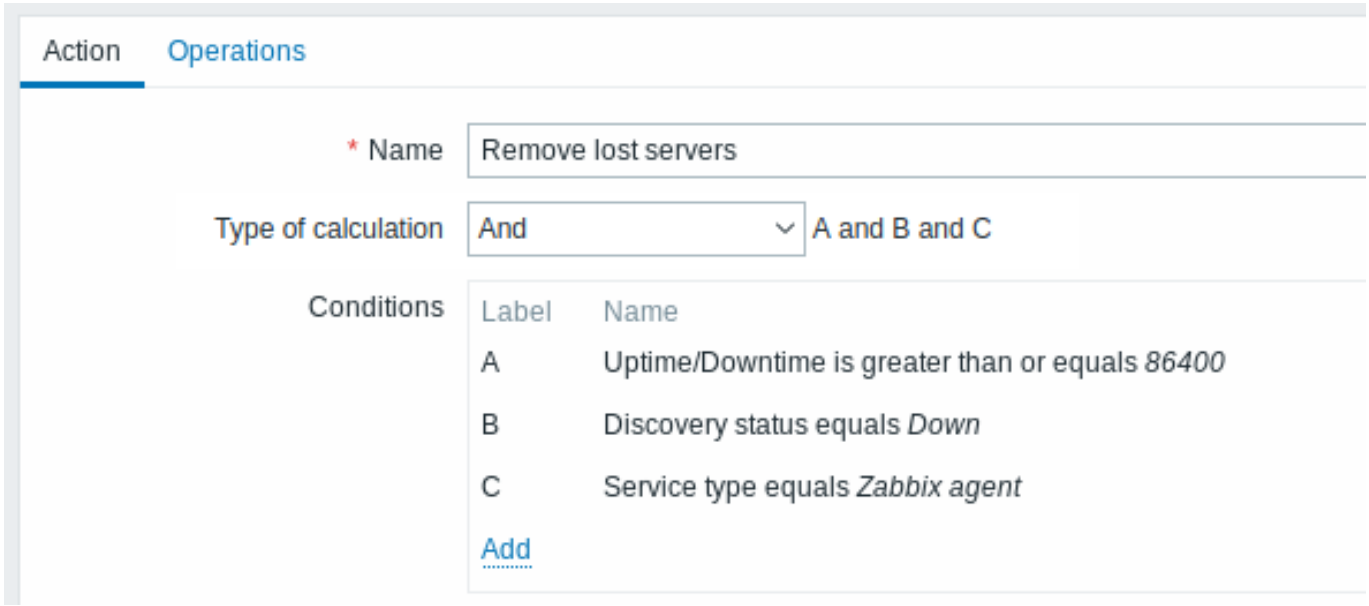
Шаг 3

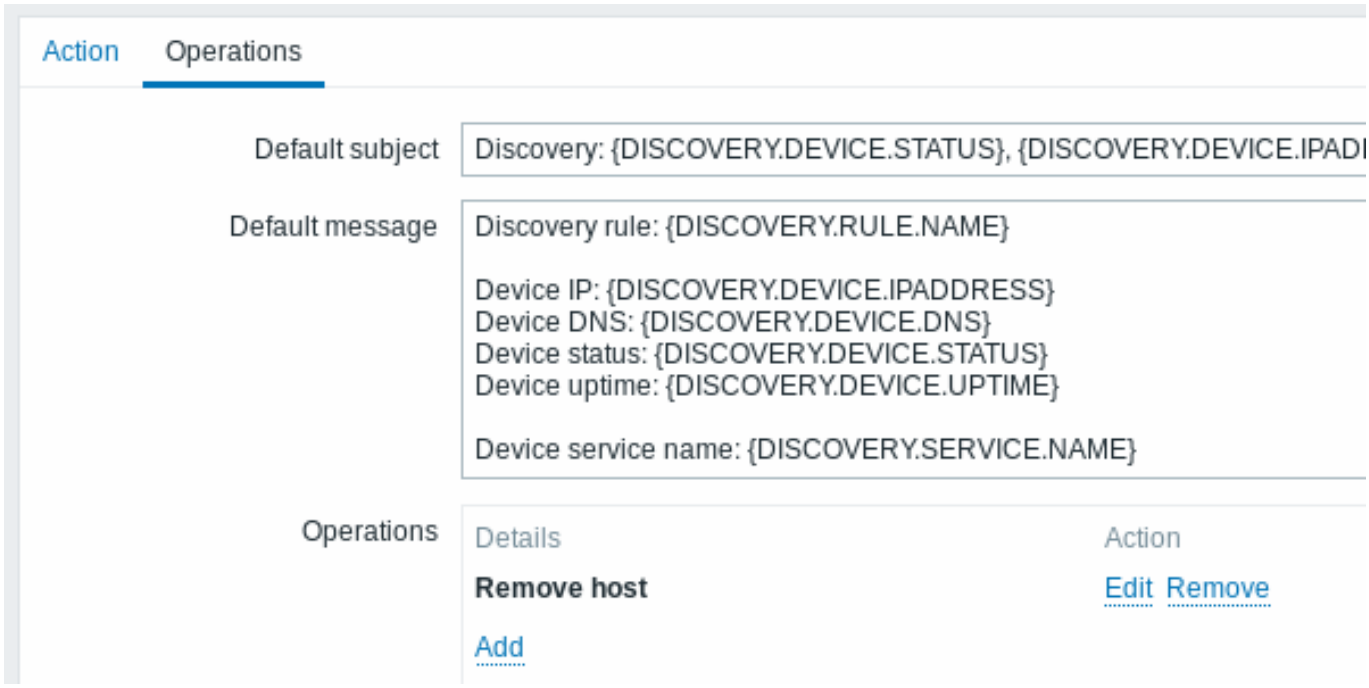
Зададим действие на обнаружение для добавления обнаруженных серверов Windows в соответствующие группы/шаблоны.



Шаг 4

Зададим действие на обнаружение для удаления потерянных серверов.





Сервер будет удалён, если сервис «Zabbix agent» в состоянии «Недоступен» на протяжении более чем 24 часов (86400 секунд).

Автоматическая регистрация активного агента

Обзор

Можно разрешить автрегистрирование активного агента Zabbix, после чего сервер сможет начать его мониторинг. Таким образом, новые узлы сети можно добавлять для мониторинга без ручной настройки на сервере.

Автрегистрация может происходить, когда ранее неизвестный активный агент запрашивает проверки.

Эта функция может быть очень полезна для автоматического мониторинга новых облачных узлов. Как только у вас появляется новый узел в Zabbix Cloud, автоматически начинается сбор данных о производительности и доступности узла сети.

Автрегистрация активного агента также поддерживает мониторинг добавленных узлов с помощью пассивных проверок. Когда активный агент запрашивает проверки, и в файле конфигурации заданы параметры `ListenIP` или `ListenPort`, они передаются на сервер. Если указано несколько IP-адресов, на сервер отправляется первый из них.

Сервер при добавлении нового автоматически зарегистрированного узла использует полученные IP-адрес и порт для настройки агента. Если значение IP-адреса не получено, используется адрес, примененный для входящего соединения.

Если значение порта не получено, используется 10050.

Можно указать, что узел сети должен быть автоматически зарегистрирован с DNS-именем в качестве интерфейса агента по умолчанию.

Автрегистрация повторно выполняется:

- если изменяется информация метаданных узла сети:
  - из-за изменения HostMetadata и перезапуска агента
  - из-за изменения значения, возвращаемого HostMetadataItem
- для созданных вручную узлов сети, у которых отсутствуют метаданные
- если вручную изменяется узел сети, чтобы он мониторился другим прокси Zabbix
- если автрегистрация для того же узла сети поступает от нового прокси Zabbix

Интервал heartbeat для автрегистрации активного агента для сервера Zabbix и прокси Zabbix составляет 120 секунд. Поэтому, если обнаруженный узел сети удален, автрегистрация будет повторно выполнена через 120 секунд.

Настройка

Укажите сервер

Убедитесь, что сервер Zabbix указан в файле конфигурации агента (`zabbix_agentd.conf`):

```
ServerActive=10.0.0.1
```

Если в `zabbix_agentd.conf` явно не задан параметр *Hostname*, сервер будет использовать системное имя узла сети, на котором работает агент, для присвоения имени узлу сети. Системное имя узла сети в Linux можно получить, выполнив команду `hostname`.

Если в конфигурации агента Zabbix параметр *Hostname* задан в виде списка узлов сети, разделённых запятыми, узлы сети будут созданы для всех указанных имён узлов сети.

После внесения изменений в файл конфигурации перезапустите агент.

Действие для автоматической регистрации активного агента

Когда сервер получает запрос на автоматическую регистрацию от агента, он вызывает действие. Для автоматической регистрации агента необходимо настроить действие с источником события «Автоматическая регистрация».

> **Note:**
> Настраивать сетевое обнаружение для автоматической регистрации активных агентов не требуется.

В веб-интерфейсе Zabbix:

1. Перейдите в раздел *Оповещения* > *Действия*.
2. Выберите *Действия автоматической регистрации*.
3. Нажмите *Создать действие*.
4. На вкладке *Действие* укажите имя действия.
5. При необходимости укажите условия. В условиях для имени узла сети или метаданных узла сети можно использовать поиск по подстроке или регулярному выражению. Если вы собираетесь использовать тип условия *Метаданные узла сети*, см. раздел Использование метаданных узла сети.
6. На вкладке *Операции* добавьте необходимые операции, например *Добавить узел сети*, *Добавить в группу узлов сети* (например, *Обнаруженные узлы сети*), *Связать шаблоны* и т. д.

> **Note:**
> Если узлы сети, которые будут автоматически регистрироваться, вероятно, поддерживают только активный мониторинг (например, узлы сети, защищенные межсетевым экраном от вашего сервера Zabbix), можно создать специальный шаблон, например *Template_Linux-active*, и связать его с узлами сети.

Созданные узлы сети добавляются в группу *Обнаруженные узлы сети* (по умолчанию; этот параметр можно изменить в разделе *Администрирование* > *Общие* > *Другие параметры*). Если вы хотите добавлять узлы сети в другую группу, добавьте операцию *Удалить из группы узлов сети* (указав «Обнаруженные узлы сети»), а также операцию *Добавить в группу узлов сети* (указав другую группу узлов сети), поскольку узел сети должен входить в группу узлов сети.

Безопасная автоматическая регистрация

Безопасная автоматическая регистрация возможна при настройке аутентификации на основе PSK с использованием зашифрованных соединений.

Уровень шифрования настраивается глобально в разделе *Администрирование* > *Общие* > *Автоматическая регистрация*. Можно выбрать отсутствие шифрования, шифрование TLS с аутентификацией PSK или оба варианта (чтобы одни узлы сети могли регистрироваться без шифрования, а другие — с использованием шифрования).

Аутентификация с помощью PSK проверяется сервером Zabbix перед добавлением узла сети. Если проверка пройдена успешно, узел сети добавляется, а параметр *Соединения с узлом сети и от узла сети* устанавливается в значение «Только PSK» с идентификатором и предварительно заданным ключом, совпадающими с указанными в глобальной настройке автоматической регистрации.

Attention:
Для обеспечения безопасности автоматической регистрации в установках, использующих прокси, следует включить шифрование между сервером Zabbix и прокси.

Использование DNS в качестве интерфейса по умолчанию

HostInterface и HostInterfaceItem параметры конфигурации позволяют указать пользовательское значение интерфейса узла сети при автоматической регистрации.

В частности, они полезны, если узел сети должен быть автоматически зарегистрирован с DNS-именем в качестве интерфейса агента по умолчанию, а не с его IP-адресом. В этом случае DNS-имя следует указать или вернуть в качестве значения одного из параметров HostInterface или HostInterfaceItem. Если значение одного из этих параметров изменяется, например с IP-адреса на DNS-имя или наоборот, интерфейс по умолчанию автоматически зарегистрированного узла сети будет соответственно обновлен. Это обновление применяется к существующему узлу сети, а не путем создания нового. Чтобы отправить новое значение, необходимо перезапустить агент, чтобы он повторно инициировал процесс автоматической регистрации.

Note:
Если параметры HostInterface или HostInterfaceItem не настроены, вместо них используется параметр listen_dns. Это значение определяется путем обратного DNS-запроса для IP-адреса агента. Если обратная DNS-зона настроена неправильно или возвращает недопустимое имя, это может привести к некорректной или неудачной автоматической регистрации из-за недопустимого значения интерфейса.

Использование метаданных узла сети

При отправке запроса на автоматическую регистрацию на сервер агент передает имя узла сети. В некоторых случаях (например, для узлов в облаке Amazon) имени узла сети недостаточно, чтобы сервер Zabbix мог различать обнаруженные узлы сети. Метаданные узла сети можно использовать дополнительно для передачи другой информации от агента на сервер.

Метаданные узла сети настраиваются в файле конфигурации агента (`zabbix_agentd.conf`). В файле конфигурации метаданные узла сети можно указать двумя способами:

```
HostMetadata
HostMetadataItem
```

Описание параметров см. по ссылке выше.

Параметр HostMetadataItem может возвращать до 65535 кодовых точек UTF-8. Более длинное значение будет усечено.

Обратите внимание, что в MySQL фактическая максимальная длина в символах будет меньше, если возвращаемое значение содержит многобайтные символы. Например, значение, содержащее только 3-байтовые символы, будет ограничено 21844 символами, а значение, содержащее только 4-байтовые символы, — 16383 символами.

Attention:
Попытка автоматической регистрации выполняется каждый раз, когда активный агент отправляет серверу запрос на обновление активных проверок. Задержка между запросами указывается в параметре RefreshActiveChecks агента. Первый запрос отправляется сразу после перезапуска агента.

Примеры

Авторегистрация по операционной системе с использованием HostMetadata

Допустим, вы хотите, чтобы узлы сети автоматически регистрировались сервером Zabbix. В вашей сети у вас есть активные агенты Zabbix (см. раздел «Настройка» выше). Также в вашей сети есть хосты Windows и хосты Linux, а в вашем веб-интерфейсе Zabbix вам доступны шаблоны «Linux by Zabbix agent» и «Windows by Zabbix agent». Поэтому вы хотите, чтобы при регистрации узла сети к регистрируемому узлу был применён соответствующий шаблон Linux/Windows. По

умолчанию, на сервер при авторегистрации отправляется только имя узла сети, чего может быть недостаточно. Чтобы убедиться, что к узлу сети применяется правильный шаблон, вы должны использовать метаданные узла сети.

Настройка в веб-интерфейсе

Первое, что надо сделать, — это настроить веб-интерфейс. Создайте 2 действия. Первое действие:

- Имя: Авторегистрация узлов сети Linux
- Условия: Метаданные узлов сети содержит *Linux*
- Операции: Присоединить шаблоны: Linux by Zabbix agent

> **Note:**
> В данном случае можно опустить операцию «Добавить узел сети». Присоединение шаблона к узлу сети требует, чтобы узел сети уже был добавлен, поэтому сервер сделает это автоматически.

Второе действие:

- Имя: Авторегистрация узлов сети Windows
- Условия: Метаданные узлов сети содержит *Windows*
- Операции: Присоединить шаблоны: Windows by Zabbix agent

Настройка агента

Теперь вам нужно настроить агентов. В файлы конфигурации агентов добавьте следующую строку:

```
HostMetadataItem=system.uname
```

Таким способом вы обеспечите, что метаданные узлов сети будут содержать строку «Linux» или «Windows» в зависимости от узла сети, на котором работает агент. Пример метаданных узла сети в этом случае:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-0PXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

Не забудьте перезапустить агента после внесения любых изменений в его файл конфигурации.

Использование HostMetadata для управления авторегистрацией и шаблонами

Шаг 1 — Защита регистрации с помощью HostMetadata

Использование метаданных узла для обеспечения некоторой базовой защиты от регистрации нежелательных узлов сети.

Настройка в веб-интерфейсе

Создайте действие в веб-интерфейсе, используя некоторый трудно-угадываемый секретный код для запрета нежелательных узлов сети:

- Имя: Действие авторегистрации Linux
- Условия:
  - Тип вычисления: И
  - Условие (A): Метаданные узлов сети содержат //Linux//
  - Условие (B): Метаданные узлов сети содержат //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae/
- Операции:
  - Отправить сообщение пользователям: Admin через все способы оповещения
  - Добавить в группы узлов сети: Linux сервера
  - Присоединить к шаблонам: Linux by Zabbix agent

Пожалуйста, обратите внимание, что использование этого метода в одиночку не предоставляет надёжную защиту, потому что данные передаются как простой текст. Для того, чтобы изменения вступили в силу немедленно, необходима перезагрузка кэша конфигурации.

Настройка агента

В файл конфигурации агента добавьте следующую строку:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

где «Linux» — это платформа, а остаток строки — трудноугадываемый секретный текст.

Не забудьте перезапустить агента после внесения любых изменений в файл конфигурации.

Step 2 — Добавление шаблона к зарегистрированному узлу сети

К уже зарегистрированному узлу сети можно добавить дополнительные шаблоны. В данном случае шаблон MySQL by Zabbix agent будет присоединён только к узлам сети, у которых HostMetadata содержит токен `MySQL`.

Настройка в веб-интерфейсе

В веб-интерфейсе обновите действие:

- Имя: Действие авторегистрации Linux
- Условия:
  - Тип вычисления: И
  - Условие (A): Метаданные узлов сети содержат Linux
  - Условие (B): Метаданные узлов сети содержат 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Условие (C): Метаданные узлов сети содержат MySQL
- Операции:
  - Отправить сообщение пользователям: Admin через все способы оповещения
  - Добавить в группы узлов сети: Linux сервера
  - Присоединить шаблоны: Linux by Zabbix agent
  - Присоединить шаблоны: MySQL by Zabbix Agent

Настройка агента

Обновите следующую строчку в конфигурационном файле агента:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Не забудьте перезапустить агента после внесения любых изменений в конфигурационный файл.

Обнаружение на низком уровне

Обзор Низкоуровневое обнаружение (LLD) автоматически обнаруживает сущности (например, файловые системы, сетевые интерфейсы) на узле сети и создает соответствующие элементы данных, триггеры и графики без ручной настройки для каждой сущности.

Чтобы использовать LLD, вы создаете **правило обнаружения**, которое собирает JSON-данные, описывающие сущности, и **прототипы** (элемент данных, триггер и график) для каждой сущности.

JSON, возвращаемый правилом, должен быть массивом объектов, где каждый объект представляет одну обнаруженную сущность с использованием пар ключ-значение. Например, правило обнаружения `net.if.discovery` может вернуть:

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Соответствующий набор элементов данных, триггеров и графиков затем создается из прототипов — по одному для каждого интерфейса (`lo` и `eth0`).

Если объекты содержат ключи в формате `{#MACRO}`, эти макросы используются напрямую в прототипах. Если нет, либо если требуются дополнительные/пользовательские макросы, макросы можно определить вручную и сопоставить с полями JSON с помощью JSONPath.

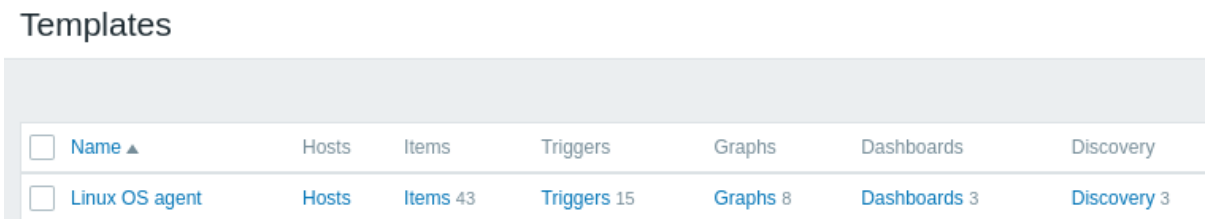
LLD также может создавать узлы сети (например, для виртуальных машин, обнаруженных на гипервизоре), и поддерживает вложенные правила обнаружения для многоуровневого обнаружения.

Сущности, которые больше не обнаруживаются, могут быть автоматически отключены или удалены.

Настройка низкоуровневого обнаружения Мы рассмотрим низкоуровневое обнаружение на примере обнаружения файловой системы.

Чтобы настроить обнаружение, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Шаблоны* или *Узлы сети*.
2. Нажмите *Обнаружение* в строке соответствующего шаблона/узла сети.



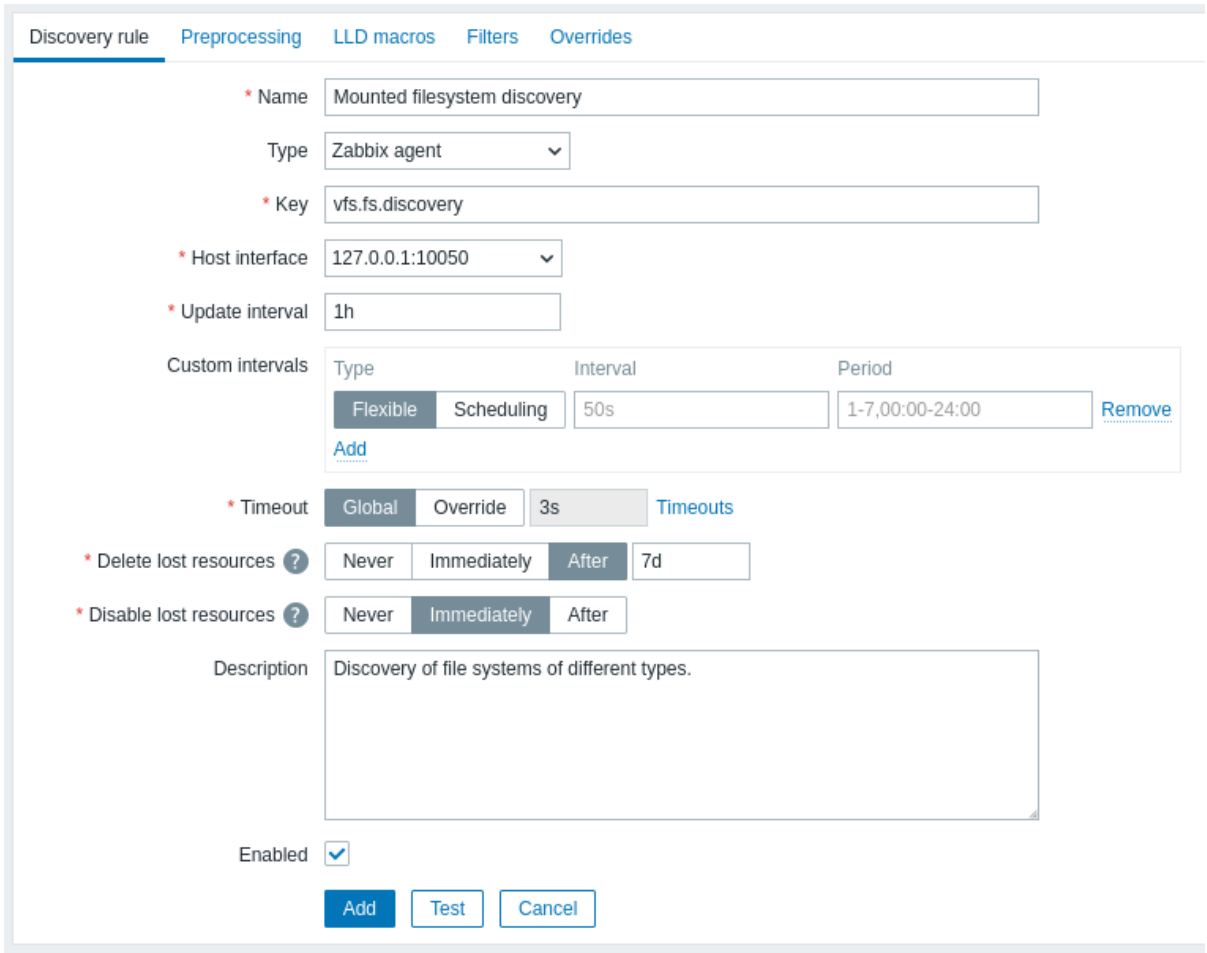
3. Нажмите *Создать правило обнаружения* в правом верхнем углу экрана.
4. Заполните форму правила обнаружения, указав необходимые сведения.

Правило обнаружения

Форма правила обнаружения содержит пять вкладок, представляющих слева направо поток данных во время обнаружения:

- *Правило обнаружения* - укажите, прежде всего, встроенный элемент данных или пользовательский скрипт для получения данных обнаружения.
- *Предобработка* - примените предобработку к обнаруженным данным.
- *Макросы LLD* - позволяют извлекать значения макросов для использования в обнаруженных элементах данных, триггерах и т. д.
- *Фильтры* - позволяют фильтровать обнаруженные значения.
- *Переопределения* - позволяют изменять элементы данных, триггеры, графики или прототипы узлов сети при применении к определенным обнаруженным объектам.

Вкладка **Правило обнаружения** содержит ключ элемента данных, используемый для обнаружения (а также некоторые общие атрибуты правила обнаружения):



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя	Имя правила обнаружения.

Параметр	Описание
Тип	Тип проверки, выполняемой для обнаружения. В этом примере используется тип элемента данных *Zabbix agent*. Правило обнаружения также может быть зависимым элементом данных, зависящим от обычного элемента данных. Оно не может зависеть от другого правила обнаружения. Для зависимого элемента данных выберите соответствующий тип (*Зависимый элемент данных*) и укажите мастер-элемент данных в поле «Мастер-элемент данных». Мастер-элемент данных должен существовать.
Ключ	Введите ключ элемента данных обнаружения (до 2048 символов). Например, можно использовать встроенный ключ элемента данных «vfs.fs.discovery», возвращающий строку JSON со списком файловых систем, присутствующих на компьютере, их типами и параметрами монтирования. Обратите внимание, что другим вариантом обнаружения файловых систем является использование результатов обнаружения, полученных с помощью ключа агента «vfs.fs.get» (см. пример).
Интервал обновления	В этом поле указывается, как часто Zabbix выполняет обнаружение. Вначале, когда вы только настраиваете обнаружение файловых систем, может потребоваться установить небольшой интервал, но после проверки работоспособности можно установить интервал в 30 минут или больше, поскольку файловые системы обычно изменяются нечасто. Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. *Примечание*: интервал обновления можно установить равным «0» только при наличии пользовательских интервалов с ненулевым значением. Если установлено значение «0» и существует пользовательский интервал (гибкий или запланированный) с ненулевым значением, опрос элемента данных будет выполняться в течение пользовательского интервала. Новые правила обнаружения будут проверены в течение 60 секунд после их создания, если для них не задан интервал обновления «Планирование» или «Гибкий» и значение *Интервал обновления* не равно 0. *Примечание*: для существующего правила обнаружения обнаружение можно выполнить немедленно, нажав кнопку *Выполнить сейчас*.
Пользовательские интервалы	Можно создать пользовательские правила проверки элемента данных: **Гибкий** - создать исключение для *Интервала обновления* (интервал с другой частотой) **Планирование** - создать пользовательское расписание опроса. Подробную информацию см. в разделе Пользовательские интервалы.
Тайм-аут	Установите тайм-аут проверки обнаружения. Выберите вариант тайм-аута: **Глобальный** - используется тайм-аут прокси/глобальный тайм-аут (отображается в неактивном поле *Тайм-аут*); **Переопределение** - используется пользовательский тайм-аут (задается в поле *Тайм-аут*; допустимый диапазон: 1 - 600s). Поддерживаются суффиксы времени, например 30s, 1m, и пользовательские макросы. Нажатие ссылки *Тайм-ауты* позволяет настроить тайм-ауты прокси или глобальные тайм-ауты (если прокси не используется). Обратите внимание, что ссылка *Тайм-ауты* видна только пользователям типа *Суперадминистратор*, имеющим права доступа к разделам веб-интерфейса *Администрирование* > *Общие* или *Администрирование* > *Прокси*.
Удаление потерянных ресурсов	Укажите, через какое время обнаруженная сущность будет удалена после изменения ее статуса обнаружения на «Больше не обнаруживается»: *Никогда* - она не будет удалена; *Немедленно* - она будет удалена немедленно; *После* - она будет удалена по истечении указанного периода времени. Значение должно быть больше значения *Отключение потерянных ресурсов*. Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы. *Примечание*: не рекомендуется использовать вариант «Немедленно», поскольку простое ошибочное редактирование фильтра может привести к удалению сущности вместе со всеми историческими данными. Обратите внимание, что ресурсы, отключенные вручную, не будут удалены обнаружением низкого уровня.

Параметр	Описание
Отключение потерянных ресурсов	Укажите, через какое время обнаруженная сущность будет отключена после изменения ее статуса обнаружения на «Больше не обнаруживается»: *Никогда* - она не будет отключена; *Немедленно* - она будет отключена немедленно; *После* - она будет отключена по истечении указанного периода времени. Значение должно быть больше интервала обновления правила обнаружения. Обратите внимание, что ресурсы, отключенные автоматически, снова станут включенными, если обнаружение низкого уровня обнаружит их повторно. Ресурсы, отключенные вручную, не будут снова включены при повторном обнаружении. Это поле не отображается, если для параметра *Удаление потерянных ресурсов* установлено значение «Немедленно». Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы.
Описание	Введите описание.
Включено	Если флажок установлен, правило будет обрабатываться.

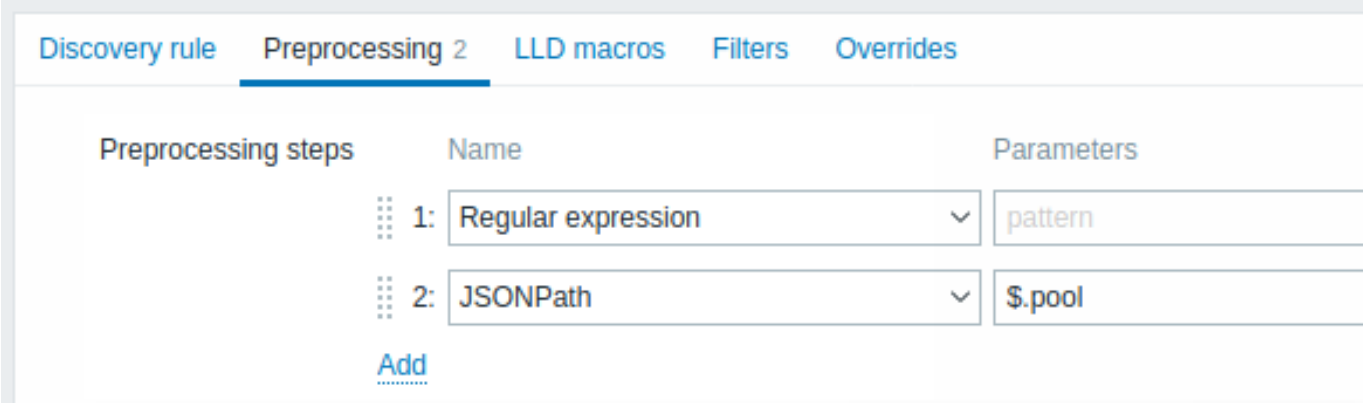
Note:
История правила обнаружения не сохраняется.

Предварительная обработка

Вкладка **Предварительная обработка** позволяет определить правила преобразования, применяемые к результату обнаружения. На этом этапе можно использовать одно или несколько преобразований. Преобразования выполняются в порядке их определения. Вся предварительная обработка выполняется сервером Zabbix.

См. также:

- Подробная информация о предварительной обработке
- Тестирование предварительной обработки



Тип	
Преобразование	Описание
Текст	
Регулярное выражение	Сопоставить полученное значение с регулярным выражением <pattern> и заменить значение извлеченным <output>. Регулярное выражение поддерживает извлечение максимум 10 захваченных групп с помощью последовательности \N. Параметры: **pattern** - регулярное выражение **output** - шаблон форматирования результата. Экранирующая последовательность \N (где N=1…9) заменяется на N-ю совпавшую группу. Экранирующая последовательность \0 заменяется на совпавший текст. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.

Тип	
Замена	Найти строку поиска и заменить ее другой строкой (или ничем). Все вхождения строки поиска будут заменены. Параметры: **search string** - строка, которую необходимо найти и заменить; учитывается регистр (обязательный параметр) **replacement** - строка, на которую необходимо заменить строку поиска. Строка замены также может быть пустой, что позволяет удалять найденную строку поиска. Для поиска или замены переводов строк, возвратов каретки, табуляций и пробелов можно использовать экранирующие последовательности "\n \r \t \s"; обратную косую черту можно экранировать как "\\"; экранирующие последовательности можно экранировать как "\\n". Экранирование переводов строк, возвратов каретки и табуляций выполняется автоматически во время обнаружения низкого уровня.
Структурированные данные	
JSONPath	Извлечь значение или фрагмент из данных JSON с помощью функциональности JSONPath. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.

XML XPath	Извлечь значение или фрагмент из данных XML с помощью функциональности XPath.
	Для работы этого параметра сервер Zabbix должен быть скомпилирован с поддержкой libxml.
	Примеры:
	`number(/document/item/value)` извлечет 10 из `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` извлечет 10 из `<document><item attribute="10"></item></document>`
	`/document/item` извлечет `<item><value>10</value></item>` из `<document><item><value>10</value></item></document>`
	Обратите внимание, что пространства имен не поддерживаются.
	Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.

CSV в JSON	Преобразовать данные файла CSV в формат JSON. Дополнительную информацию см. в разделе Предварительная обработка CSV в JSON.
XML в JSON	Преобразовать данные в формате XML в JSON. Дополнительную информацию см. в разделе Правила сериализации. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
SNMP	
Значение SNMP walk	Извлечь значение по указанному имени OID/MIB и применить параметры форматирования: **Без изменений** - вернуть Hex-STRING в виде неэкранированной шестнадцатеричной строки (*примечание*: подсказки отображения по-прежнему применяются); **UTF-8 из Hex-STRING** - преобразовать Hex-STRING в строку UTF-8; **MAC из Hex-STRING** - преобразовать Hex-STRING в строку MAC-адреса (при этом `' '` будет заменен на `':'`); **Целое число из BITS** - преобразовать первые 8 байт битовой строки, представленной в виде последовательности шестнадцатеричных символов (например, "1A 2B 3C 4D"), в 64-битное беззнаковое целое число. В битовых строках длиной более 8 байт последующие байты игнорируются. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.

Тип	
SNMP walk в JSON	Преобразовать значения SNMP в JSON. Укажите имя поля в JSON и соответствующий путь OID SNMP. Значения полей будут заполнены значениями из указанного пути OID SNMP. Этот этап предварительной обработки можно использовать для обнаружения OID SNMP. Доступны аналогичные параметры форматирования значений, что и на этапе *Значение SNMP walk*. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Значение SNMP get	Применить параметры форматирования к значению SNMP get: **UTF-8 из Hex-STRING** - преобразовать Hex-STRING в строку UTF-8; **MAC из Hex-STRING** - преобразовать Hex-STRING в строку MAC-адреса (при этом `' '` будет заменен на `':'`); **Целое число из BITS** - преобразовать первые 8 байт битовой строки, представленной в виде последовательности шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное беззнаковое целое число. В битовых строках длиной более 8 байт последующие байты игнорируются. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Пользовательские скрипты	
JavaScript	Введите код JavaScript в модальном редакторе, который открывается при нажатии в поле параметра или на значке карандаша рядом с ним. Обратите внимание, что допустимая длина JavaScript зависит от используемой базы данных. Дополнительную информацию см. в разделе Предварительная обработка JavaScript
Проверка	
Не соответствует регулярному выражению	Укажите регулярное выражение, которому значение не должно соответствовать. Например, `Error:(.*?)\.` Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Проверка наличия ошибки в JSON	Проверить наличие сообщения об ошибке на уровне приложения, расположенного по пути JSONPath. Остановить обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжить обработку со значением, которое было до этого этапа предварительной обработки. Обратите внимание, что эти ошибки внешней службы сообщаются пользователю как есть, без добавления информации об этапе предварительной обработки. Например, `$.errors`. Если получен JSON вида `{"errors":"e1"}`, следующий этап предварительной обработки выполнен не будет. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Проверка наличия ошибки в XML	Проверить наличие сообщения об ошибке на уровне приложения, расположенного по пути XPath. Остановить обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжить обработку со значением, которое было до этого этапа предварительной обработки. Обратите внимание, что эти ошибки внешней службы сообщаются пользователю как есть, без добавления информации об этапе предварительной обработки. При ошибке разбора некорректного XML ошибка сообщаться не будет. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Соответствует регулярному выражению	Укажите регулярное выражение, которому должно соответствовать значение. Если установлен флажок *Настраиваемое действие при ошибке*, можно указать настраиваемые параметры обработки ошибок: отбросить значение, задать указанное значение или задать указанное сообщение об ошибке.
Ограничение частоты	

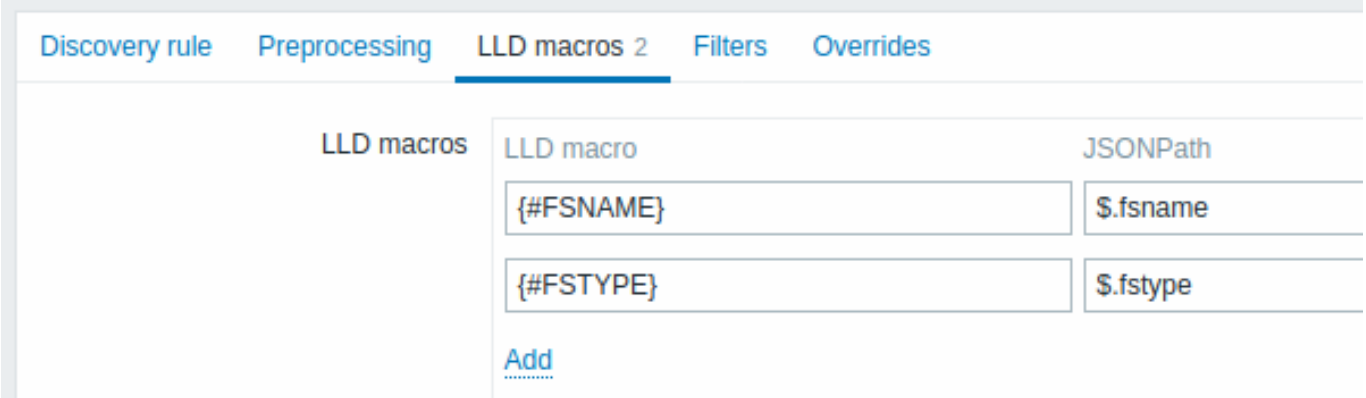
Тип	
Отбрасывать неизменившиеся значения с контрольным интервалом	Отбрасывать значение, если оно не изменялось в течение заданного периода времени (в секундах). Для указания количества секунд поддерживаются положительные целые числа (минимум - 1 секунда). В этом поле можно использовать суффиксы времени (например, `30s`, `1m`, `2h`, `1d`). В этом поле можно использовать пользовательские макросы и макросы обнаружения низкого уровня. Для элемента данных обнаружения можно указать только один параметр ограничения частоты. Например, `1m`. Если одинаковый текст передается этому правилу дважды в течение 60 секунд, он будет отброшен. *Примечание*: изменение прототипов элементов данных не сбрасывает ограничение частоты. Ограничение частоты сбрасывается только при изменении этапов предварительной обработки.
Prometheus	
Prometheus в JSON	Преобразовать необходимые метрики Prometheus в JSON. Дополнительную информацию см. в разделе Проверки Prometheus.

Обратите внимание, что если правило обнаружения применено к узлу сети через шаблон, содержимое этой вкладки доступно только для чтения.

Пользовательские макросы

Вкладка **LLD macros** позволяет указать пользовательские LLD macros, если возвращаемый JSON еще не содержит ключей в формате `{#MACRO}`.

Например, ключ элемента данных агента `vfs.fs.get` возвращает JSON с данными файловой системы без каких-либо предопределенных LLD macros.
Здесь вы вручную определяете макросы и сопоставляете их со значениями в JSON с помощью выражений JSONPath:



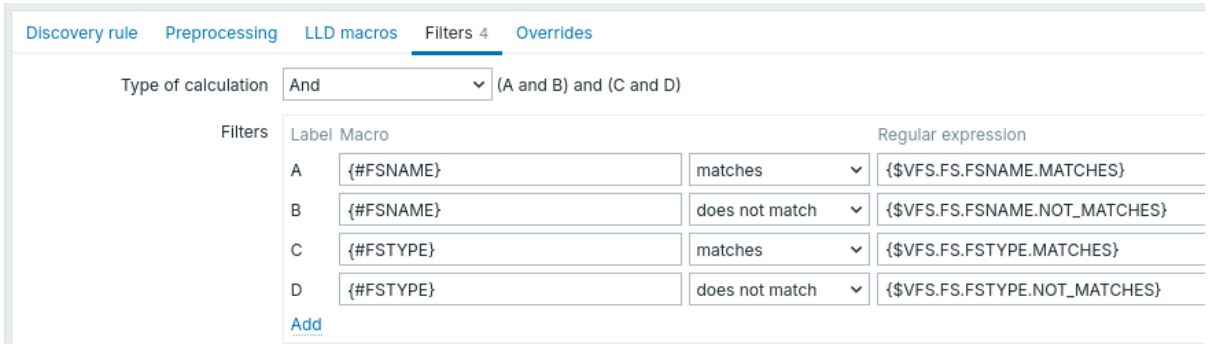
Parameter	Description
LLD macro	Имя LLD macro в формате `{#MACRO}`.
JSONPath	Выражение, которое определяет, какое значение извлекать из каждой обнаруженной сущности в массиве JSON. Например, `$.foo` извлекает `"bar"` и `"baz"` из `[{"foo":"bar"}, {"foo":"baz"}]`. JSONPath поддерживает как точечную, так и скобочную нотацию. Скобочная нотация должна использоваться для ключей, содержащих специальные символы или Unicode, например, `$['unicode + special chars #1']['unicode + special chars #2']`.

Правило обнаружения должно возвращать массив JSON в корне (`$.`).
Для обратной совместимости также допускается объект JSON, содержащий единственный массив `"data"`. В этом случае массив `"data"` автоматически разворачивается:

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Фильтр

Фильтр можно использовать для создания реальных элементов данных, триггеров и графиков только для сущностей, соответствующих заданным критериям. Вкладка **Фильтры** содержит определения фильтров правил обнаружения, которые позволяют фильтровать значения обнаружения:



Параметр	Описание
Тип вычисления	Доступны следующие варианты вычисления фильтров: **И** - должны пройти все фильтры; **Или** - достаточно прохождения одного фильтра; **И/Или** - использует *И* для разных имен макросов и *Или* для одинаковых имен макросов; **Пользовательское выражение** - позволяет задать пользовательское вычисление фильтров. Формула должна включать все фильтры из списка. Ограничение - 255 символов.
Фильтры	Доступны следующие операторы условий фильтра: *соответствует*, *не соответствует*, *существует*, *не существует*. Операторы *соответствует* и *не соответствует* требуют регулярное выражение, совместимое с Perl (PCRE). В поле *Регулярное выражение* можно ввести регулярное выражение или указать глобальное регулярное выражение. Операторы *существует* и *не существует* позволяют фильтровать сущности на основе наличия или отсутствия указанного макроса LLD в ответе. Обратите внимание: если макрос из фильтра отсутствует в ответе, найденная сущность будет проигнорирована, если для этого макроса не указано условие «не существует». Если отсутствие макроса влияет на результат выражения, будет отображено предупреждение. Например, если `{#B}` отсутствует в выражении: `{#A}` соответствует 1 и `{#B}` соответствует 2 - будет отображено предупреждение `{#A}` соответствует 1 или `{#B}` соответствует 2 - предупреждение не отображается.

Например, если вас интересуют только файловые системы C:, D: и E:, укажите `{#FSNAME}` в поле *Макрос*, а `^C|^D|^E` - в поле *Регулярное выражение*.
 Фильтрация также возможна по типам файловых систем с помощью макроса `{#FSTYPE}` (например, `^ext|^reiserfs`) и по типам дисков (поддерживается только агентом Windows) с помощью макроса `{#FSDRIVETYPE}` (например, `fixed`).

Для проверки регулярного выражения можно использовать `grep -E`, например:

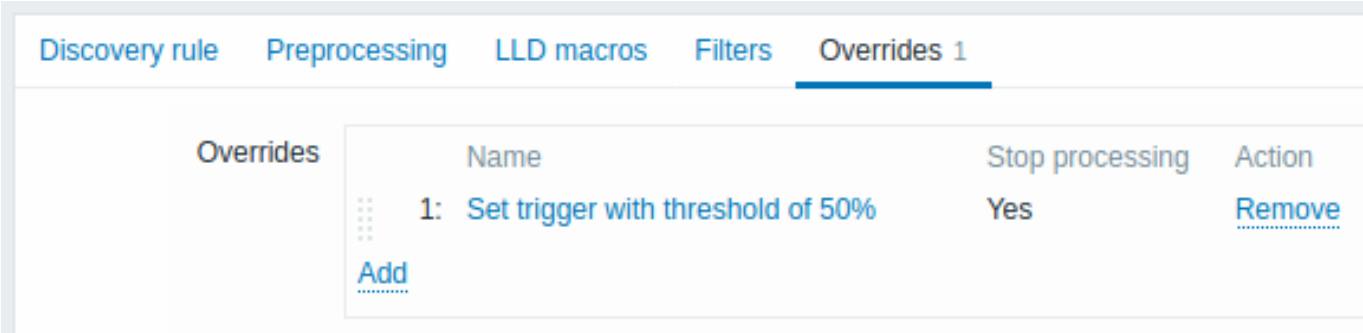
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Ошибка или опечатка в регулярном выражении, используемом в правиле LLD (например, некорректное регулярное выражение «Файловые системы для обнаружения»), может привести к удалению тысяч элементов конфигурации, исторических значений и событий для множества узлов сети.

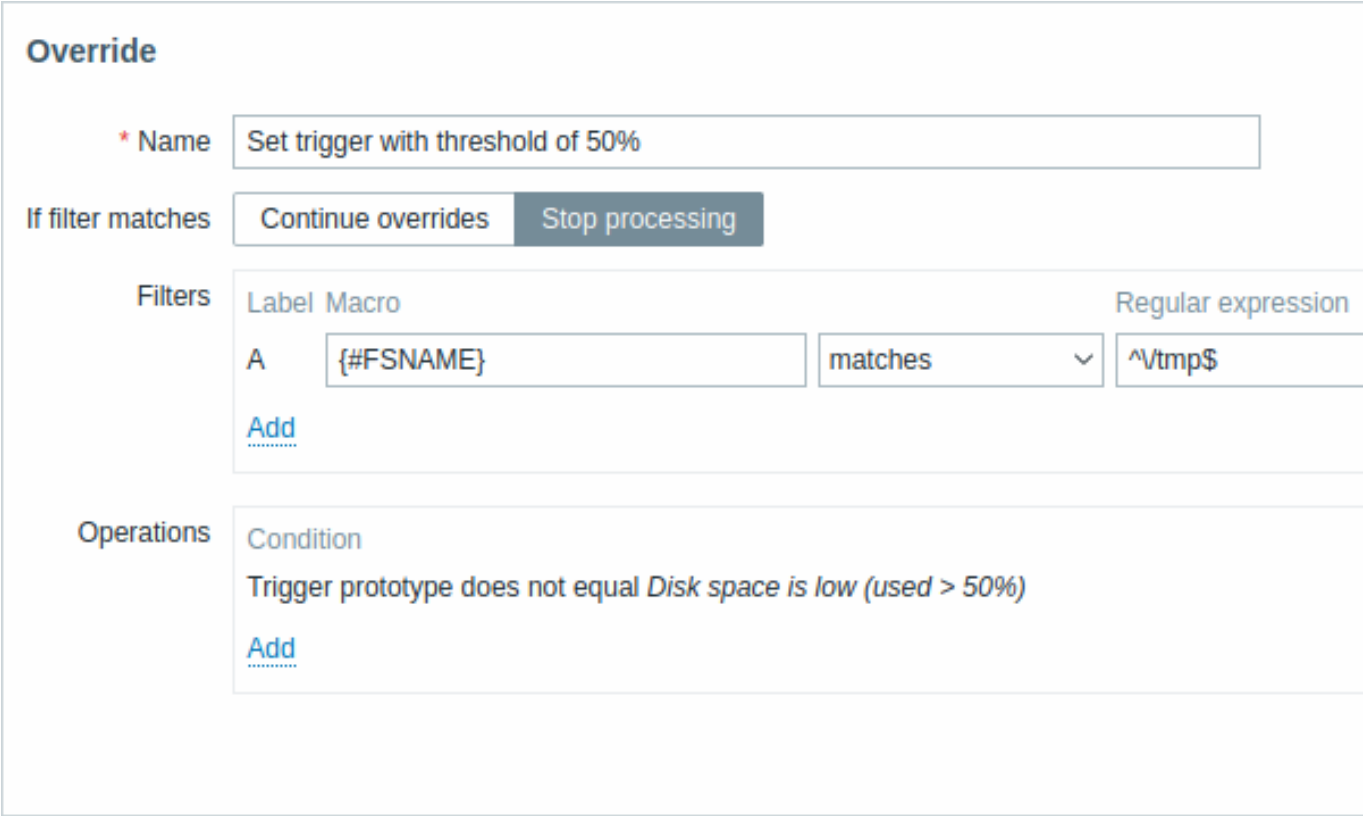
Attention:
База данных Zabbix в MySQL должна быть создана с учетом регистра, чтобы имена файловых систем, различающиеся только регистром, обнаруживались корректно.

Переопределение

Вкладка **Переопределения** позволяет задавать правила для изменения списка прототипов элементов данных, триггеров, графиков, узлов сети и обнаружения либо их атрибутов для обнаруженных объектов, соответствующих заданным критериям.



Переопределения (если они есть) отображаются в списке с возможностью изменения порядка методом перетаскивания и выполняются в порядке их определения. Чтобы настроить параметры нового переопределения, нажмите Add в блоке *Переопределения*. Чтобы изменить существующее переопределение, нажмите его имя. Откроется всплывающее окно, в котором можно изменить параметры правила переопределения.

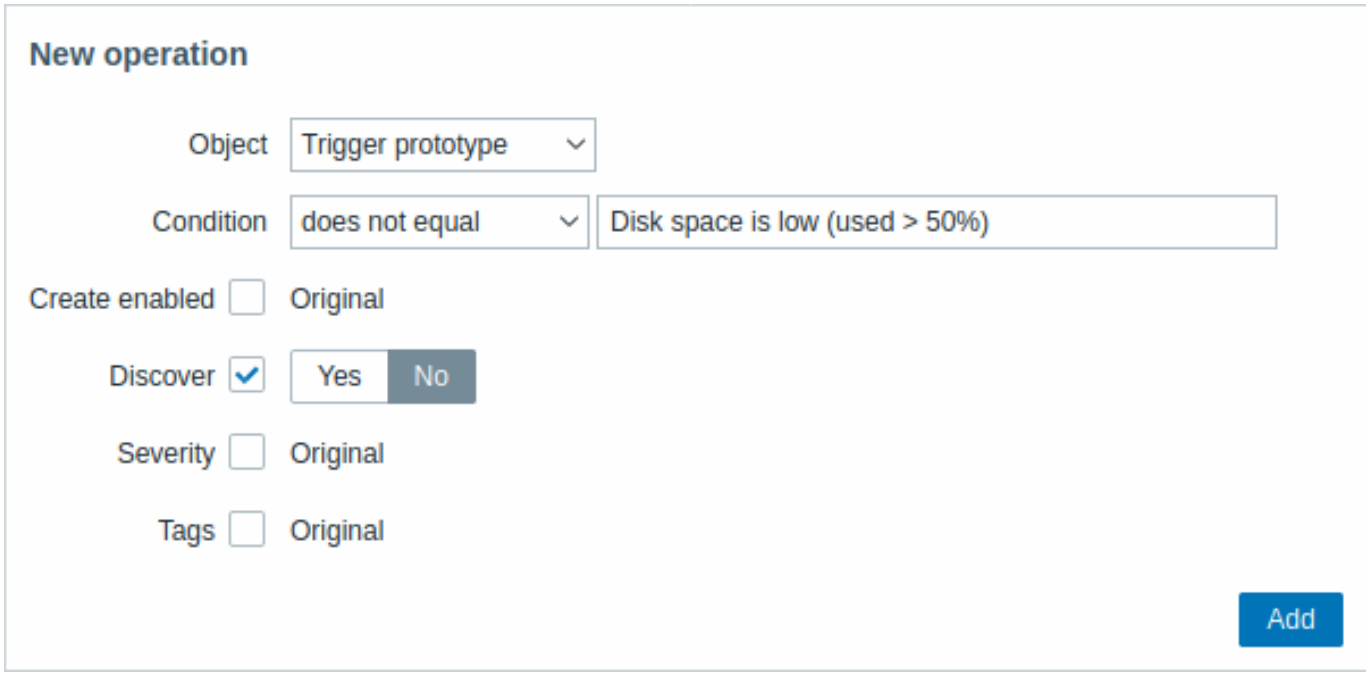


Все обязательные параметры отмечены красными звездочками.

Параметр	Описание
Имя	Уникальное (в рамках правила LLD) имя переопределения.
Если фильтр совпал	Определяет, следует ли обрабатывать следующие переопределения при выполнении условий фильтра: **Продолжить переопределения** - последующие переопределения будут обработаны. **Остановить обработку** - будут выполнены операции предыдущих переопределений (если есть) и этого переопределения, а последующие переопределения будут проигнорированы для совпавших строк LLD.
Фильтры	Определяет, к каким обнаруженным сущностям следует применить переопределение. Фильтры переопределения обрабатываются после фильтров правила обнаружения и имеют те же функции.
Операции	Операции переопределения отображаются со следующими сведениями: **Условие** - тип объекта и условие, которому должно соответствовать имя объекта; например: прототип триггера не равен *Мало места на диске (использовано > 50%)*. **Действия** - отображаются ссылки для изменения и удаления операции.

Настройка операции

Чтобы настроить параметры новой операции, нажмите Add в блоке «Операции». Чтобы изменить существующую операцию, нажмите Edit рядом с операцией. Откроется всплывающее окно, в котором можно изменить параметры операции.



Параметр	Описание
Объект	Доступны пять типов объектов: Прототип элемента данных Прототип триггера Прототип графика Прототип узла сети Прототип обнаружения
Условие	Позволяет фильтровать сущности, к которым следует применить операцию.
Оператор	Поддерживаемые операторы: **равно** - применить к этому прототипу **не равно** - применить ко всем прототипам, кроме этого **содержит** - применить, если имя прототипа содержит эту строку **не содержит** - применить, если имя прототипа не содержит эту строку **соответствует** - применить, если имя прототипа соответствует регулярному выражению **не соответствует** - применить, если имя прототипа не соответствует регулярному выражению
Шаблон	Регулярное выражение или строка для поиска.
Объект: *Прототип элемента данных*	
Создать включенным	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа элемента данных: *Да* - элемент данных будет добавлен во включенном состоянии. *Нет* - элемент данных будет добавлен в обнаруженную сущность, но в отключенном состоянии.
Обнаруживать	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа элемента данных: *Да* - элемент данных будет добавлен. *Нет* - элемент данных не будет добавлен.

Параметр	Описание
Интервал обновления	Если флажок установлен, появятся два параметра, позволяющие задать другой интервал для элемента данных: *Задержка* - интервал обновления элемента данных. Поддерживаются пользовательские макросы, макросы LLD и суффиксы времени (например, 30s, 1m, 2h, 1d). Следует установить значение 0, если используется *Пользовательский интервал*. *Пользовательский интервал* - нажмите Add, чтобы указать гибкие интервалы или интервалы по расписанию. Подробную информацию см. в разделе Пользовательские интервалы.
История	Если флажок установлен, появятся кнопки, позволяющие задать другие периоды хранения истории для элемента данных: *Не хранить* - если выбрано, история не будет храниться. *Хранить до* - если выбрано, справа появится поле для указания периода хранения. Поддерживаются пользовательские макросы и макросы LLD.
Тренды	Если флажок установлен, появятся кнопки, позволяющие задать другие периоды хранения трендов для элемента данных: *Не хранить* - если выбрано, тренды не будут храниться. *Хранить до* - если выбрано, справа появится поле для указания периода хранения. Поддерживаются пользовательские макросы и макросы LLD.
Теги	Если флажок установлен, появится новый блок, позволяющий указать пары «тег-значение». Поддерживаются пользовательские макросы и макросы LLD. Эти теги будут добавлены к тегам, указанным в прототипе элемента данных, даже если имена тегов совпадают.
Объект: *Прототип триггера*	
Создать включенным	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа триггера: *Да* - триггер будет добавлен во включенном состоянии. *Нет* - триггер будет добавлен в обнаруженную сущность, но в отключенном состоянии.
Обнаруживать	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа триггера: *Да* - триггер будет добавлен. *Нет* - триггер не будет добавлен.
Важность	Если флажок установлен, появятся кнопки важности триггера, позволяющие изменить важность триггера.
Теги	Если флажок установлен, появится новый блок, позволяющий указать пары «тег-значение». Поддерживаются пользовательские макросы и макросы LLD. Эти теги будут добавлены к тегам, указанным в прототипе триггера, даже если имена тегов совпадают.
Объект: *Прототип графика*	
Обнаруживать	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа графика: *Да* - график будет добавлен. *Нет* - график не будет добавлен.
Объект: *Прототип узла сети*	
Создать включенным	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа узла сети: *Да* - узел сети будет создан во включенном состоянии. *Нет* - узел сети будет создан в отключенном состоянии.
Обнаруживать	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа узла сети: *Да* - узел сети будет обнаружен. *Нет* - узел сети не будет обнаружен.

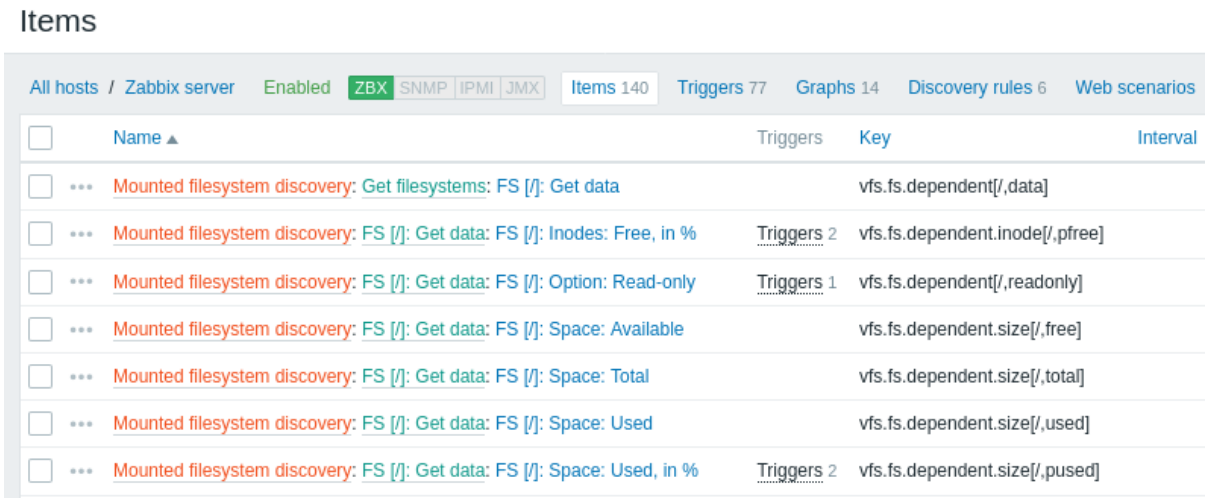
Параметр	Описание
Связать шаблоны	Если флажок установлен, появится поле для указания шаблонов. Начните вводить имя шаблона или нажмите *Выбрать* рядом с полем и выберите шаблоны из списка во всплывающем окне. Шаблоны из этого переопределения добавляются ко всем шаблонам, уже связанным с прототипом узла сети.
Теги	Если флажок установлен, появится новый блок, позволяющий указать пары «тег-значение». Поддерживаются пользовательские макросы и макросы LLD. Эти теги будут добавлены к тегам, указанным в прототипе узла сети, даже если имена тегов совпадают.
Инвентаризация узла сети	Если флажок установлен, появятся кнопки, позволяющие выбрать другой режим инвентаризации для прототипа узла сети: *Отключена* - не заполнять инвентаризацию узла сети *Вручную* - вводить сведения вручную *Автоматически* - автоматически заполнять данные инвентаризации узла сети на основе собранных метрик.
Объект: *Прототип обнаружения*	
Создать включенным	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа обнаружения: *Да* - правило обнаружения будет добавлено во включенном состоянии. *Нет* - правило обнаружения будет добавлено в отключенном состоянии.
Обнаруживать	Если флажок установлен, появятся кнопки, позволяющие переопределить исходные настройки прототипа обнаружения: *Да* - правило обнаружения будет добавлено. *Нет* - правило обнаружения не будет добавлено.
Интервал обновления	Если флажок установлен, появятся два параметра, позволяющие задать другой интервал для правила обнаружения: *Задержка* - интервал обновления правила. Поддерживаются пользовательские макросы, макросы LLD и суффиксы времени (например, 30s, 1m, 2h, 1d). Следует установить значение 0, если используется *Пользовательский интервал*. *Пользовательский интервал* - нажмите Add, чтобы указать гибкие интервалы или интервалы по расписанию. Подробную информацию см. в разделе Пользовательские интервалы.

Кнопки формы

Кнопки в нижней части формы позволяют выполнить несколько операций.

Add	Добавить правило обнаружения. Эта кнопка доступна только для новых правил обнаружения.
Update	Обновить свойства правила обнаружения. Эта кнопка доступна только для существующих правил обнаружения.
Clone	Создать другое правило обнаружения на основе свойств текущего правила обнаружения.
Execute now	Немедленно выполнить обнаружение на основе правила обнаружения. Правило обнаружения уже должно существовать. См. подробнее. *Примечание*: при немедленном выполнении обнаружения кэш конфигурации не обновляется, поэтому результат не будет отражать самые последние изменения конфигурации правила обнаружения.
Test	Проверить конфигурацию правила обнаружения. Используйте эту кнопку, чтобы проверить параметры конфигурации (например, подключение и корректность параметров) без окончательного применения каких-либо изменений.
Delete	Удалить правило обнаружения.
Cancel	Отменить редактирование свойств правила обнаружения.

Обнаруженные объекты Представленные ниже снимки экрана иллюстрируют, как выглядят уже обнаруженные элементы данных, триггеры и графики в настройках узла сети. Обнаруженные объекты имеют префикс — ссылку золотистого цвета, которая ведёт к правилу обнаружения, создавшему эти объекты.



Обратите внимание, что обнаруженные объекты не будут созданы в случае, если объекты с такими же условиями уникальности уже существуют — например, элемент данных с таким же ключом или график с таким же именем. В таком случае в веб-интерфейсе отобразится сообщение об ошибке, что правило низкоуровневого обнаружения не смогло создать определённые объекты. Само правило обнаружения, тем не менее, не станет неподдерживаемым из-за того, что некоторые объекты не смогли быть созданы и были пропущены. Правило обнаружения перейдёт к созданию/обновлению других объектов.

Если обнаруженный объект (узел сети, файловая система, интерфейс и т.д.) более не обнаруживается (или более не попадает под фильтр), то объекты, созданные на его основе, могут быть автоматически деактивированы и впоследствии удалены.

Потерянные ресурсы могут быть автоматически деактивированы на основе значения параметра *Удаление потерянных ресурсов (Delete lost resources)*. Это влияет на потерянные узлы сети, группы узлов сети, элементы данных, триггеры и графики.

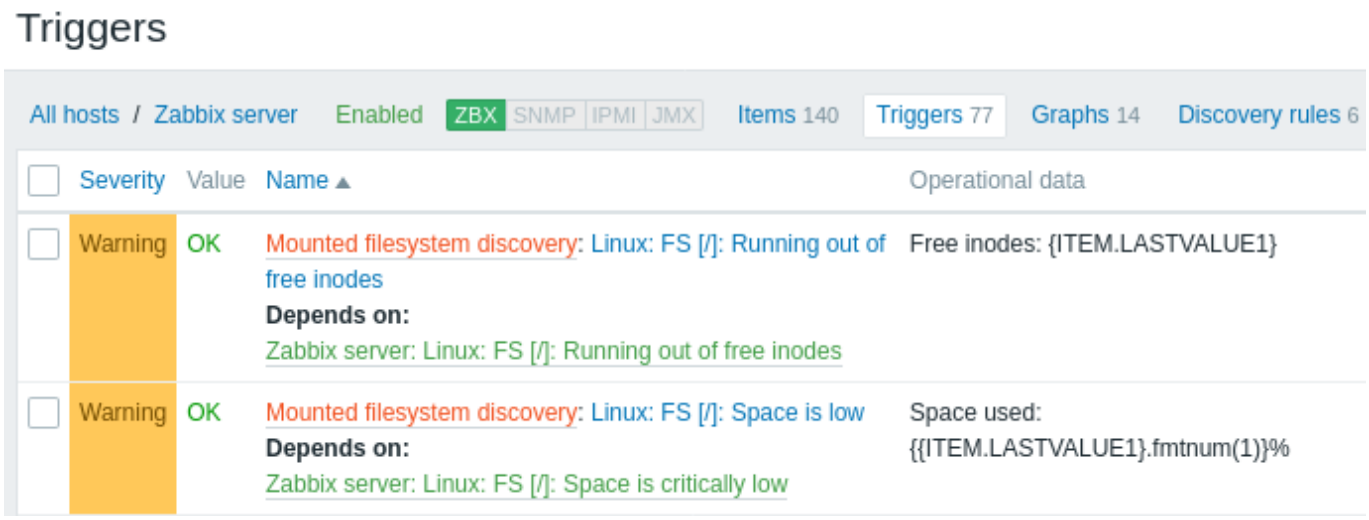
Потерянные ресурсы могут быть автоматически удалены на основе значения параметра *Деактивация потерянных ресурсов (Disable lost resources)*. Это влияет на потерянные узлы сети, элементы данных и триггеры.

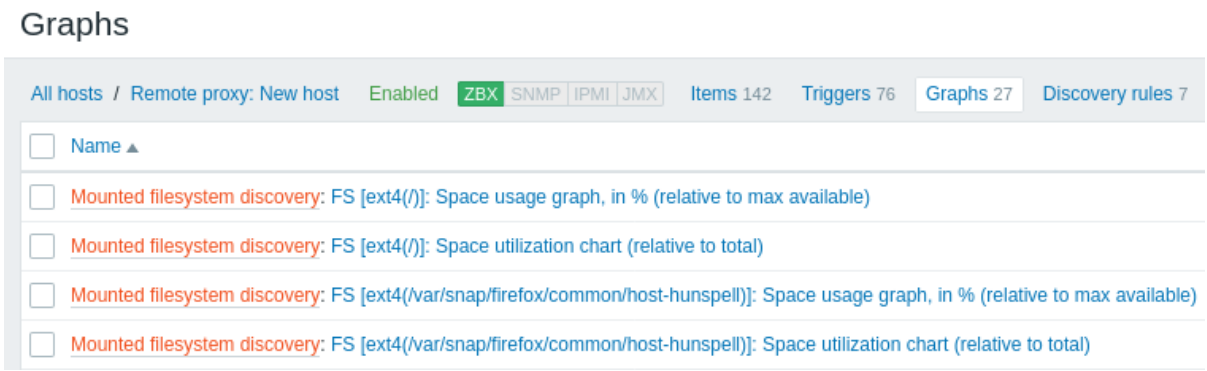
Когда обнаруженный объект становится «Более не обнаруживается», в списке элементов данных будет отображаться оранжевый индикатор времени жизни. Переместите курсор мыши на этот индикатор, и вы увидите сообщение с количеством дней до момента удаления элемента данных.



Если объекты помечены на удаление, но не были удалены в назначенное время (деактивировано правило обнаружения или узел сети элемента данных), они удалятся при следующем выполнении правила обнаружения.

Объекты, которые содержат другие объекты, которые помечены на удаление, не будут обновлены, если будут изменены на уровне правила обнаружения. Например, триггеры на основе LLD не будут обновлены, если они содержат элементы данных, которые помечены на удаление.





Другие типы обнаружения Для получения более подробных сведений и инструкций по остальным типам доступных «из коробки» обнаружений смотрите следующие разделы:

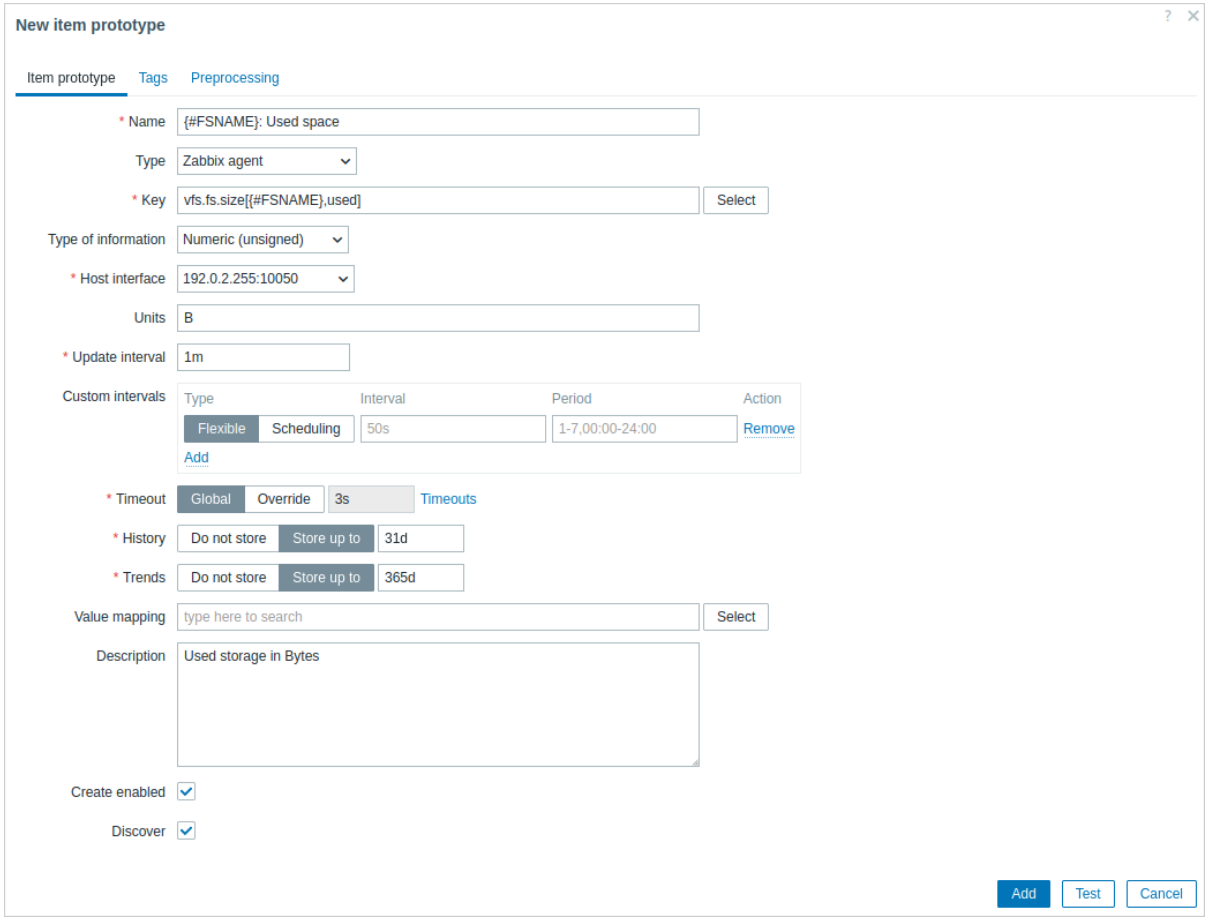
- обнаружение сетевых интерфейсов;
- обнаружение CPU и ядер CPU;
- обнаружение SNMP OID-ов;
- обнаружение объектов JMX;
- обнаружение с использованием запросов ODBC SQL;
- обнаружение служб Windows;
- обнаружение интерфейсов узлов сети в Zabbix.

Для получения более подробных сведений касательно JSON формата по обнаружению элементов данных и примера, каким образом реализовать своё собственное обнаружение файловых систем при помощи Perl скрипта, смотрите создание пользовательских правил LLD.

Прототипы элементов данных

После создания правила перейдите к элементам данных этого правила и нажмите *Создать прототип элемента данных*, чтобы создать прототип элемента данных.

Обратите внимание, как используется макрос `{#FSNAME}` там, где требуется имя файловой системы. Использование макроса обнаружения низкого уровня в ключе элемента данных обязательно, чтобы обеспечить корректную обработку обнаружения. При обработке правила обнаружения этот макрос будет заменен обнаруженной файловой системой.



Note:
Если параметры ключа элемента данных не используются, поместите макрос LLD внутри скобок ключа элемента данных `[...]` (например, `v_[{#MACRO}]`).

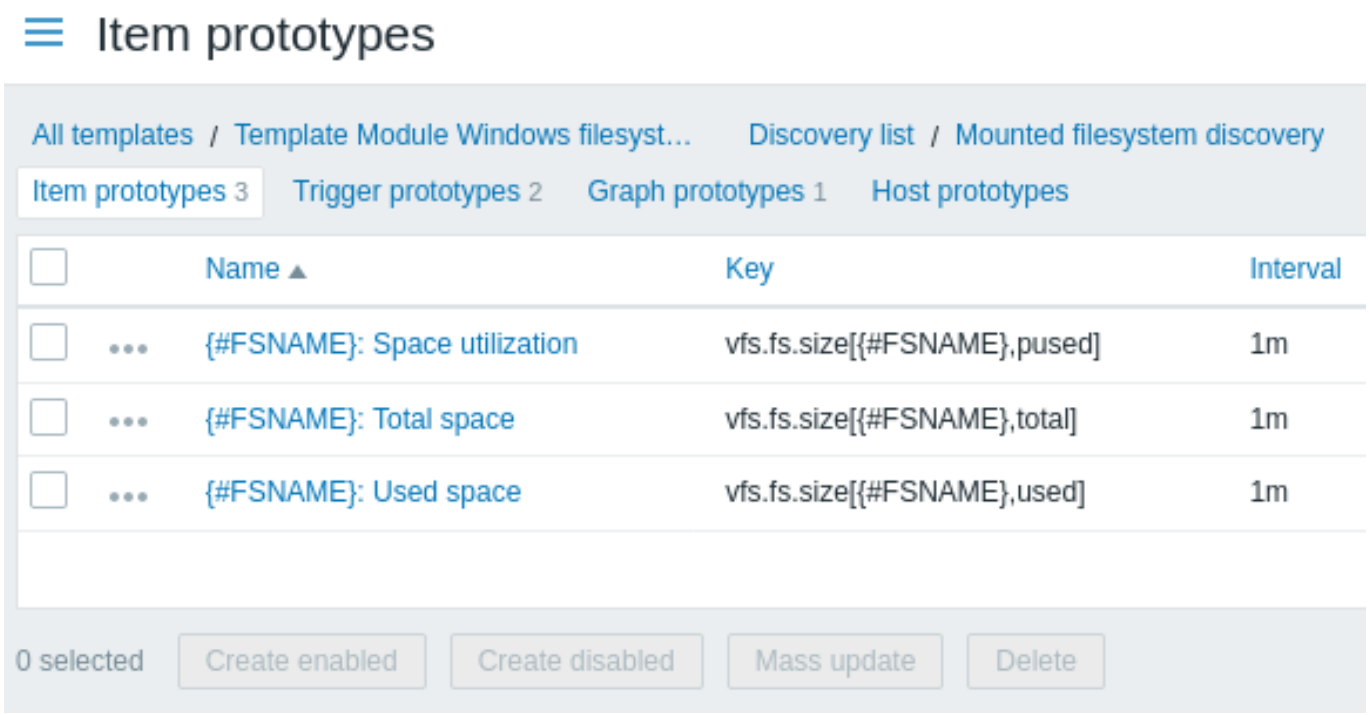
Макросы обнаружения низкого уровня и пользовательские макросы поддерживаются в настройках прототипа элемента данных и параметрах предварительной обработки значений элемента данных. Обратите внимание: при использовании в интервалах обновления один макрос должен заполнять всё поле. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются.

Note:
Для безопасного использования в параметрах предварительной обработки регулярных выражений и XPath выполняется контекстное экранирование макросов обнаружения низкого уровня.

Атрибуты, специфичные для прототипов элементов данных:

Параметр	Описание
Создавать включенным	Если флажок установлен, элемент данных будет добавлен во включенном состоянии. Если флажок не установлен, элемент данных будет добавлен в обнаруженную сущность, но в отключенном состоянии.
Обнаруживать	Если флажок установлен (по умолчанию), элемент данных будет добавлен в обнаруженную сущность. Если флажок не установлен, элемент данных не будет добавлен в обнаруженную сущность, если только этот параметр не будет переопределен в правиле обнаружения.

Для каждой интересующей нас метрики файловой системы можно создать несколько прототипов элементов данных:



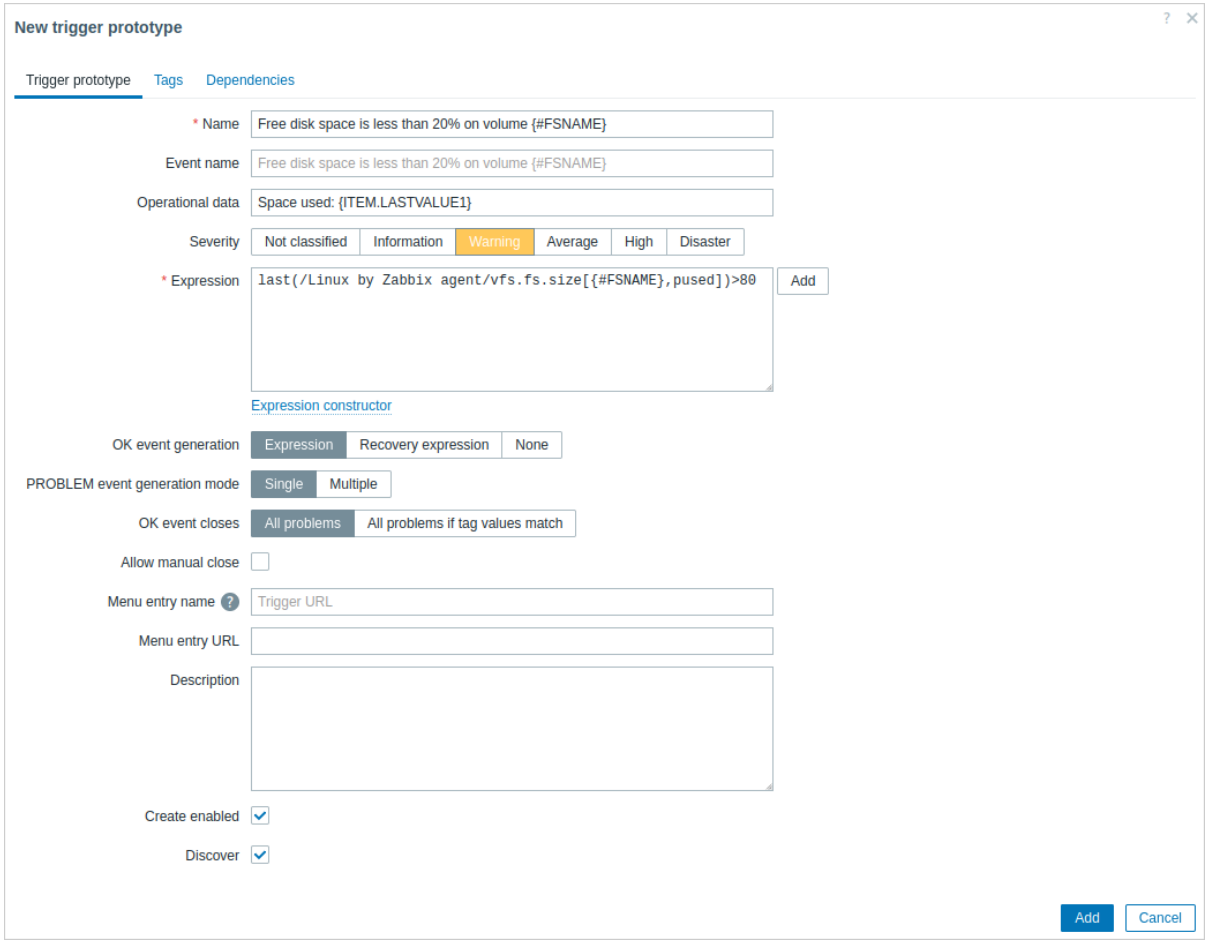
Нажмите значок с тремя точками, чтобы открыть меню конкретного прототипа элемента данных со следующими параметрами:

- *Создать прототип триггера* — создать прототип триггера на основе этого прототипа элемента данных
- *Прототипы триггеров* — нажмите, чтобы просмотреть список ссылок на уже настроенные прототипы триггеров этого прототипа элемента данных
- *Создать зависимый элемент данных* — создать зависимый элемент данных для этого прототипа элемента данных

Параметр *Массовое обновление* доступен, если требуется одновременно обновить свойства нескольких прототипов элементов данных.

Прототипы триггеров

Мы создаем прототипы триггеров аналогично прототипам элементов данных:

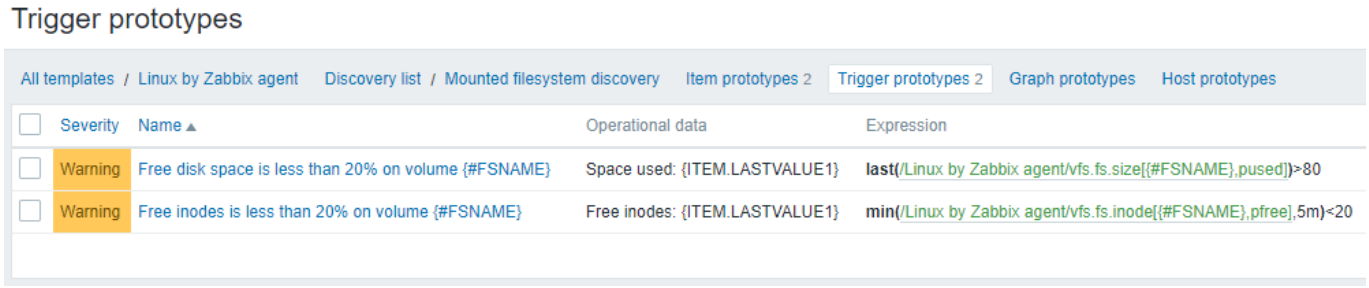


Атрибуты, специфичные для прототипов триггеров:

Parameter	Description
Create enabled	Если флажок установлен, триггер будет добавлен в включенном состоянии. Если флажок снят, триггер будет добавлен к обнаруженному объекту, но в отключенном состоянии.
Discover	Если флажок установлен (по умолчанию), триггер будет добавлен к обнаруженному объекту. Если флажок снят, триггер не будет добавлен к обнаруженному объекту, если только этот параметр не будет переопределен в правиле обнаружения.

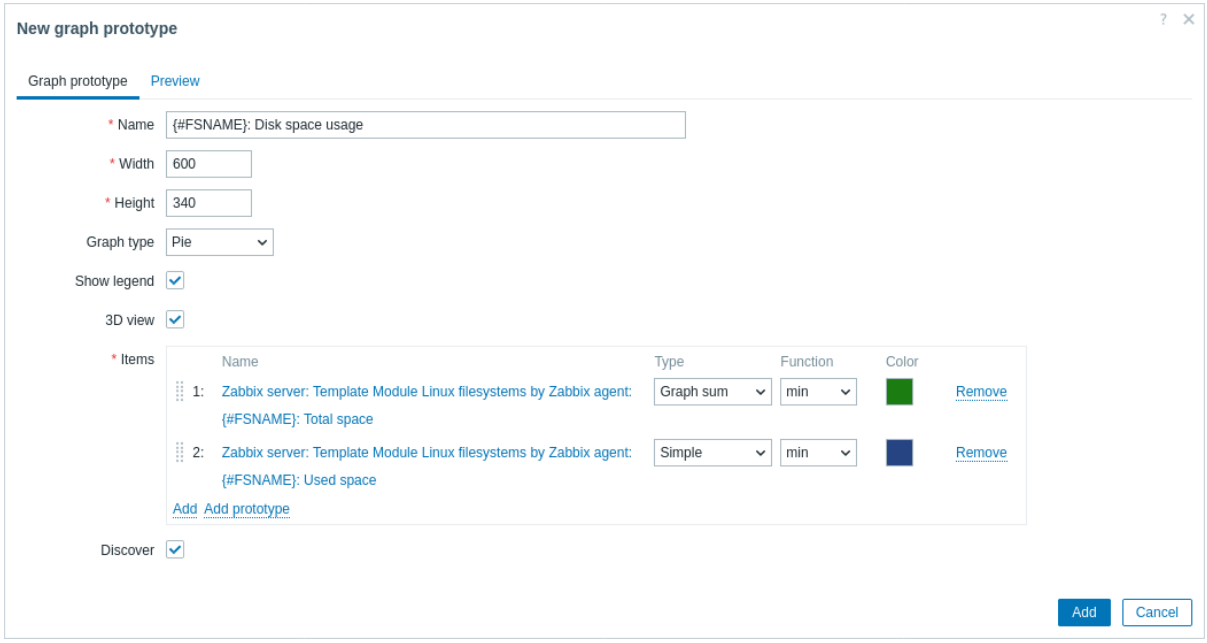
Когда реальные триггеры создаются из прототипов, может возникнуть необходимость гибко выбирать, какая константа ('20' в нашем примере) используется для сравнения в выражении. См. как пользовательские макросы с контекстом могут быть полезны для реализации такой гибкости.

Вы можете задать зависимости между прототипами триггеров. Для этого перейдите на вкладку *Dependencies*. Прототип триггера может зависеть от другого прототипа триггера из того же правила обнаружения на низком уровне (LLD) или от обычного триггера. Прототип триггера не может зависеть от прототипа триггера из другого правила LLD или от триггера, созданного из прототипа триггера. Прототип триггера узла сети не может зависеть от триггера из шаблона.



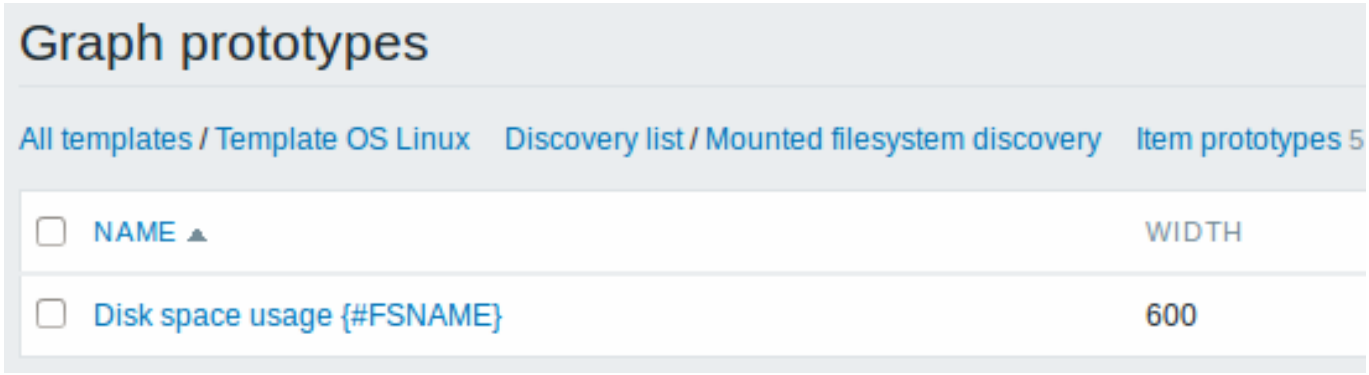
Прототипы графиков

Мы также можем создавать прототипы графиков:

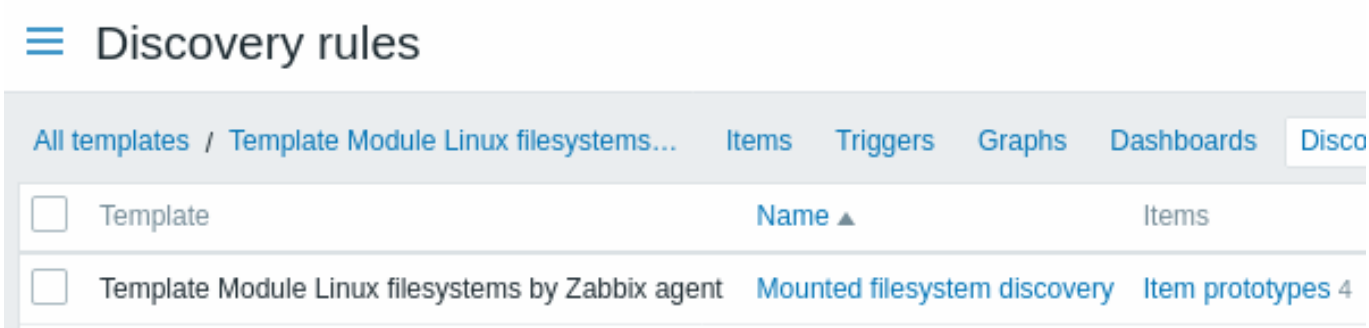


Атрибуты, специфичные для прототипов графиков:

Parameter	Description
Discover	Если флажок установлен (по умолчанию), график будет добавлен к обнаруженному объекту. Если флажок снят, график не будет добавлен к обнаруженному объекту, если только этот параметр не будет переопределен в правиле обнаружения.



Наконец, мы создали правило обнаружения, которое выглядит так, как показано ниже. Оно имеет пять прототипов элементов данных, два прототипа триггеров и один прототип графика.



Прототипы узлов сети

Прототипы узлов сети представляют собой заготовки для создания узлов сети с помощью правил низкоуровневого обнаружения. До того, как отработает обнаружение и будут созданы узлы сети, эти прототипы не могут иметь элементов данных и триггеров, за исключением тех, которые присоединены из шаблонов.

Конфигурация

Прототипы узлов сети настраиваются в правилах низкоуровневого обнаружения.

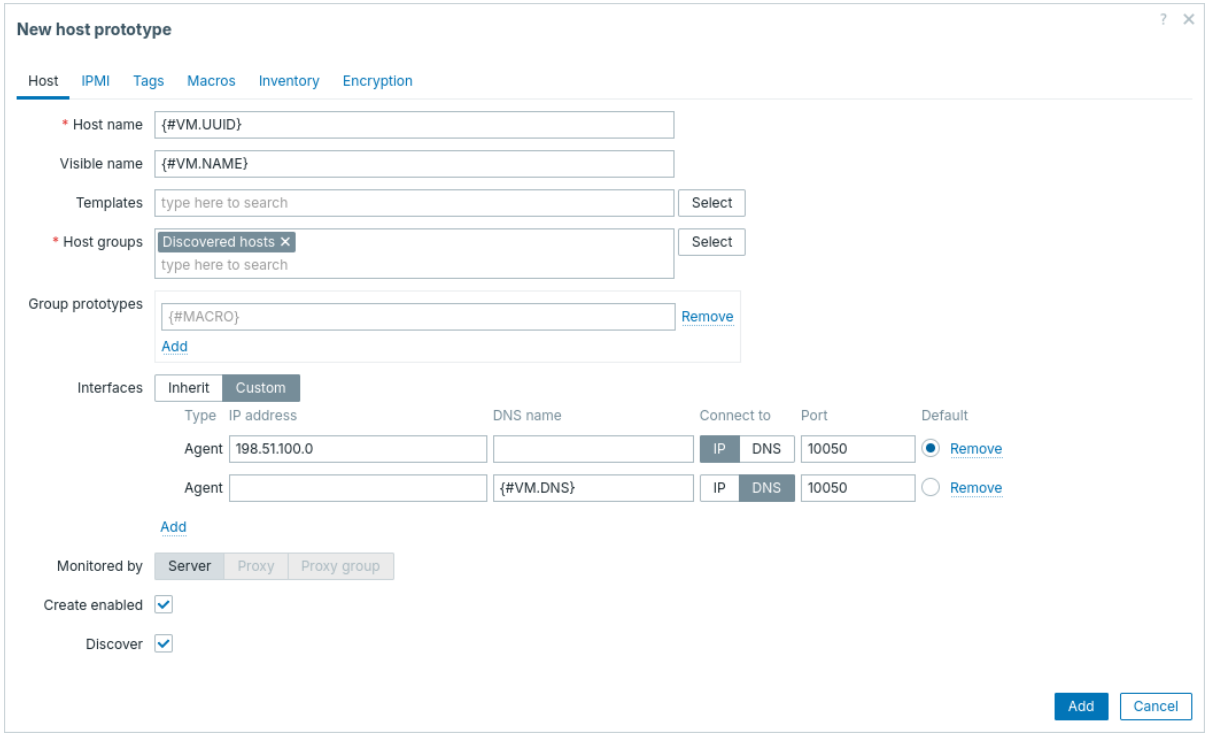
Чтобы создать прототип узла сети:

1. Перейдите в *Сбор данных > Узлы сети*.

2. Нажмите *Обнаружение* для нужного узла сети, чтобы перейти к списку правил низкоуровневого обнаружения, настроенных для этого узла сети.

3. Нажмите *Прототипы узлов сети* для нужного правила обнаружения.

4. Нажмите кнопку *Создать прототип узла сети* в правом верхнем углу и заполните форму настройки.



Все обязательные поля ввода отмечены красной звездочкой.

Parameter	Description
Host name	Этот параметр должен содержать как минимум один макрос низкоуровневого обнаружения, чтобы обеспечить уникальные имена узлов сети для создаваемых узлов сети.
Visible name	Поддерживаются макросы низкоуровневого обнаружения.
Group prototypes	Позволяет указывать прототипы групп узлов сети с использованием макросов низкоуровневого обнаружения. На основе указанных прототипов групп будут обнаружены, созданы и привязаны к создаваемым узлам сети группы узлов сети; также будут привязаны обнаруженные группы, которые уже были созданы другими правилами низкоуровневого обнаружения. Однако обнаруженные группы узлов сети, совпадающие с вручную созданными группами узлов сети, не будут привязаны к создаваемым узлам сети.
Interfaces	Укажите, должны ли обнаруженные узлы сети наследовать IP от узла сети, которому принадлежит правило обнаружения (по умолчанию), или получать пользовательские интерфейсы. Поддерживаются макросы низкоуровневого обнаружения и пользовательские макросы.
Monitored by	Это поле автоматически синхронизируется с соответствующим полем в конфигурации узла сети.
Create enabled	Укажите статус обнаруженных узлов сети; если флажок не установлен, узлы сети будут созданы как отключенные.
Discover	Укажите, будут ли узлы сети создаваться из прототипа узла сети; если флажок не установлен, узлы сети не будут создаваться из прототипа узла сети (если только эта настройка не будет переопределена в правиле низкоуровневого обнаружения).

Note:
Макросы низкоуровневого обнаружения также поддерживаются для значений тегов и значений пользовательских макросов прототипа узла сети.
 Карты значений не поддерживаются для прототипов узлов сети.

Пример настройки прототипа узла сети см. в разделе *Мониторинг виртуальных машин*.

Интерфейсы узла сети

Чтобы добавить пользовательские интерфейсы, переключите селектор *Интерфейсы (Interfaces)* из положения «Наследовать (Inherit)» в «Пользовательское (Custom)». Нажмите Add (Добавить) и выберите тип интерфейса — Zabbix агент, SNMP, JMX или IPMI.

Note:
Если выбрано *Пользовательское (Custom)*, но интерфейс не указан, узел сети будет создан без интерфейсов.
 Если выбрано *Наследовать (Inherit)*, а прототип узла сети принадлежит шаблону, то все обнаруженные узлы сети унаследуют интерфейс узла сети от того узла сети, к которому присоединён шаблон.

Если указано несколько пользовательских интерфейсов, то основной интерфейс можно задать в столбце *По умолчанию (Default)*.

Для примера того, как настраивать пользовательский интерфейс узла сети, смотрите *Пример настройки мониторинга VMware*.

Warning:
Узел сети будет создан, только если интерфейс узла сети содержит корректные данные.

Обнаруженные узлы сети

В списке узлов сети обнаруженные узлы сети помечаются префиксом с именем правила обнаружения, которое их создало.

Обнаруженные узлы сети наследуют большинство параметров от прототипов узлов сети как *только для чтения*. Настраивать можно только следующие параметры обнаруженных узлов сети:

- *Шаблоны* - привязывать новые шаблоны или отвязывать добавленные вручную. Шаблоны, унаследованные от прототипа узла сети, отвязать нельзя.
- *Описание* - добавлять или редактировать описание узла сети.
- *Статус* - включать или отключать узел сети.
- *Теги* - добавлять новые теги или удалять добавленные вручную. Теги, унаследованные от прототипа узла сети, удалить нельзя. Обратите внимание, что теги не могут иметь дубликатов (теги с одинаковыми именем и значением). Если в прототип узла сети добавлен новый тег и он совпадает с тегом, добавленным вручную на обнаруженном узле сети, то при обнаружении вручную добавленный тег будет заменен унаследованным тегом.
- *Макросы* - добавлять новые макросы узла сети или удалять добавленные вручную, а также унаследованные от прототипа узла сети; изменять значения макросов и типы.

Note:
Обнаруженные узлы сети также наследуют пользовательские макросы от узла сети, на котором настроено правило обнаружения. Эти унаследованные макросы можно удалять, а также изменять их значения и типы.

Обнаруженные узлы сети могут также обнаруживать другие узлы сети. Например, если у прототипа узла сети (используемого для обнаружения гипервизоров) есть привязанный к нему шаблон, который включает правило обнаружения низкого уровня со своим собственным прототипом узла сети (используемым для обнаружения виртуальных машин), Zabbix обнаружит и гипервизоры, и их виртуальные машины. Чтобы дополнительно обнаруживать контейнеры на этих виртуальных машинах, можно создать правила обнаружения низкого уровня или привязать новый шаблон с прототипами узлов сети к обнаруженным виртуальным машинам либо предварительно настроить прототип узла сети для виртуальных машин, привязав шаблон, который сам включает прототипы узлов сети.

Обнаруженные узлы сети можно удалять вручную. Однако обратите внимание, что они будут обнаружены снова, если для них включено обнаружение.

Узлы сети, которые больше не обнаруживаются, могут быть:

- автоматически отключены (на основе значения *Отключать потерянные ресурсы* в правиле обнаружения)
- автоматически удалены (на основе значения *Удалять потерянные ресурсы* в правиле обнаружения).

Прототипы обнаружения

Обзор

Прототипы обнаружения — это **вложенные правила обнаружения низкого уровня** внутри «родительского» правила обнаружения, позволяющие выполнять многоуровневое обнаружение объектов с собственными элементами данных, триггерами и т. д. Например, можно обнаружить все экземпляры баз данных на сервере баз данных, затем обнаружить табличные пространства для каждого экземпляра, а затем таблицы для каждого табличного пространства.

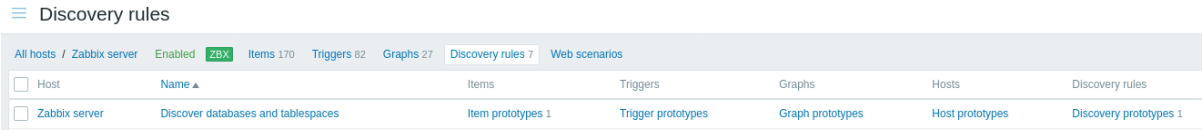
Прототипы обнаружения имеют собственные прототипы элементов данных, триггеров, графиков, узлов сети и обнаружения. Вложенный прототип обнаружения будет использовать то же значение JSON, что и родительское правило, если указать тип *Nested*.

Количество уровней вложенности прототипов обнаружения не ограничено.

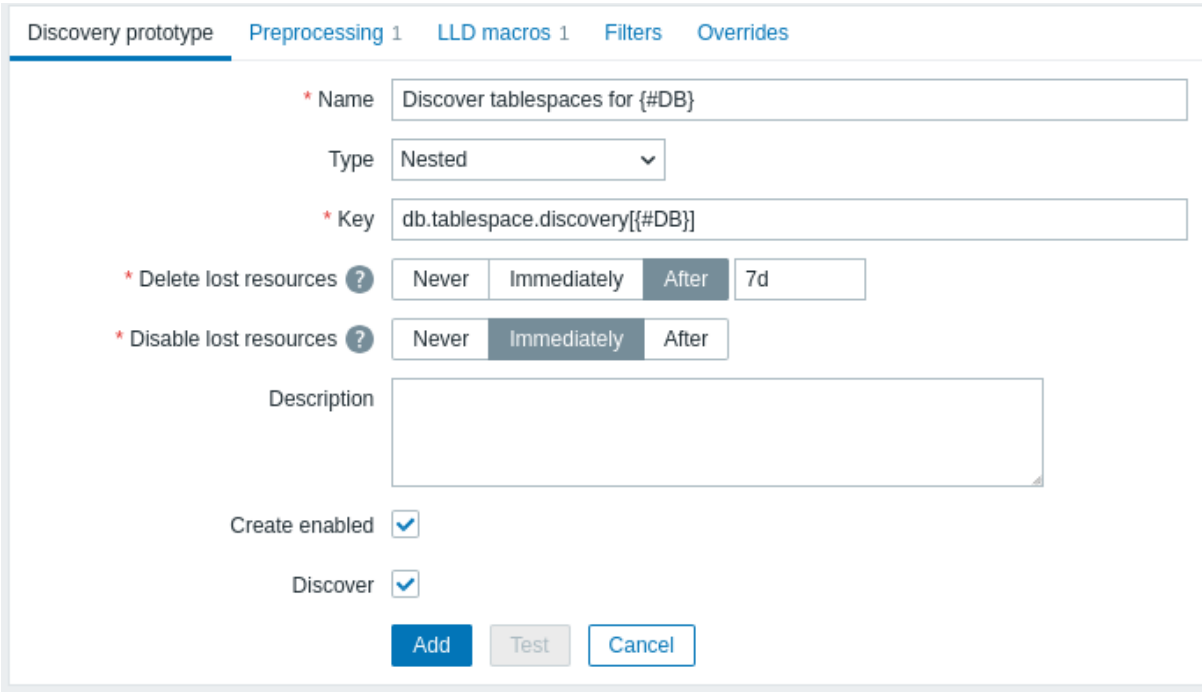
Конфигурация

Чтобы создать прототип обнаружения:

- Нажмите *Прототипы обнаружения* в строке существующего правила обнаружения



- Нажмите *Создать прототип обнаружения*



Поля конфигурации этой формы общие с обычным обнаружением низкого уровня.

Если в открывшейся форме прототипа обнаружения в качестве *Type* выбрать ”Nested”, то правила обнаружения (из прототипа обнаружения) будут сгенерированы на основе объекта JSON из того же значения JSON, что и у родительского правила обнаружения. Например, если исходный JSON — [<object A>, <object B>] и существует один вложенный прототип правила обнаружения, то будут сгенерированы два правила обнаружения на основе данных object A и object B соответственно.

В этом случае прототип обнаружения активируется одновременно с родительским правилом. Таким образом, вложенное правило может использовать предварительную обработку, чтобы работать с другим ”срезом” тех же данных, уже полученных родительским правилом.

LLD-макросы из родительского LLD-правила доступны для вложенных правил обнаружения.

Вложенные правила LLD на обнаруженных узлах сети

Вложенное правило низкоуровневого обнаружения можно использовать в шаблоне узла сети, назначенном прототипу узла сети. Если *вложенное* правило обнаружения существует на обнаруженном узле сети, тогда объект JSON,

используемый для обнаружения узла сети, также отправляется всем правилам LLD вложенного типа на этом узле сети. Подробнее см. в примере.

Макросы LLD из правила обнаружения, создавшего узел сети, доступны для вложенных правил обнаружения.

Пример

Давайте проиллюстрируем возможное применение прототипов обнаружения на основе получения следующего примера многоуровневого JSON.

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
  },
  {
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```

Случай 1

Обнаружение экземпляров баз данных на сервере баз данных, затем обнаружение табличных пространств для каждого экземпляра.

1. У вас есть как минимум один узел сети, связанный с обнаружением сервера баз данных.
2. Создайте правило LLD для этого узла сети с именем *Обнаружение баз данных и табличных пространств*.
3. Перейдите на вкладку *LLD Macros* этого правила и добавьте макрос `{#DB}=$.database`.
4. Добавьте прототип элемента данных для этого правила с именем *Активные подключения к {#DB}* (Тип: Агент, ключ: `db.connections[{#DB}]`).
5. Обнаруживаются элементы данных, относящиеся к каждой базе данных:

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. Создайте прототип обнаружения для этого правила с именем *Обнаружение табличных пространств для {#DB}* (Тип: Nested, ключ: `db.tablespace.discovery[{#DB}]`).
7. Перейдите на вкладку *Preprocessing* этого прототипа обнаружения и добавьте шаг `JSONPath=$.tablespaces`.
8. Перейдите на вкладку *LLD Macros* этого прототипа обнаружения и добавьте макрос `{#TSNAME}=$.name`.
9. Создайте прототип элемента данных для этого прототипа обнаружения с именем *Размер табличного пространства {#TSNAME} для {#DB}* (Тип: Агент, ключ: `db.ts.size[{#DB}, {#TSNAME}]`).

10. Обнаруживаются элементы данных, относящиеся к каждому табличному пространству каждой базы данных:

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

с ключами `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Случай 2

Обнаружение экземпляров базы данных на сервере базы данных с представлением их как обнаруженных узлов сети, а затем обнаружение табличных пространств для каждого экземпляра.

1. У вас есть как минимум один узел сети (корневой узел сети), связанный с обнаружением сервера базы данных.
2. Создайте шаблон для обнаружения табличных пространств для каждой базы данных.
3. Создайте в этом шаблоне элемент данных с именем *Активные подключения к {#DB}* (Тип: Агент, ключ: `db.connections[{#DB}]`).
4. Создайте правило LLD для этого шаблона с именем *Обнаружить табличные пространства* (Тип: Вложенный).
5. Перейдите на вкладку *Предобработка* этого правила и добавьте шаг `JSONPath=$.tablespaces`.
6. Перейдите на вкладку *Макросы LLD* этого правила, добавьте макрос `{#TSNAME}=$.name`.
7. Создайте прототип элемента данных для этого правила с именем *Размер табличного пространства {#TSNAME} для {#DB}* (Тип: Агент, ключ: `db.ts.size[{#DB}, {#TSNAME}]`).
8. Вернувшись к корневому узлу сети, создайте правило LLD для этого узла сети с именем *Обнаружить базы данных и табличные пространства*.
9. Перейдите на вкладку *Макросы LLD* этого правила, добавьте макрос `{#DB}=$.database`.
10. Добавьте прототип узла сети для этого правила с именем *Узел сети для базы данных {#DB}*.
11. Перейдите на вкладку *Макросы* этого прототипа узла сети, добавьте макрос `{$DB}={#DB}` (для имени и ключа элемента данных из шага 3).
12. Привяжите шаблон из шага 2 к этому прототипу узла сети.
13. Обнаруженные узлы сети содержат обнаруженные элементы данных, относящиеся к каждой базе данных и её табличным пространствам:

Узел сети	Элементы данных
Узел сети для базы данных db1	Активные подключения к db1
	Размер табличного пространства ts1 для db1
	Размер табличного пространства ts2 для db1
	Размер табличного пространства ts3 для db1
Узел сети для базы данных db2	Активные подключения к db2
	Размер табличного пространства ts1 для db2
	Размер табличного пространства ts2 для db2
	Размер табличного пространства ts3 для db2
Узел сети для базы данных db3	Активные подключения к db3
	Размер табличного пространства ts1 для db3
	Размер табличного пространства ts2 для db3
	Размер табличного пространства ts3 для db3

Примечания по низкоуровневому обнаружению

Использование LLD макросов в контекстах пользовательских макросов

LLD макросы могут использоваться внутри контекста пользовательского макроса, например, в прототипах триггера.

Несколько правил низкоуровневого обнаружения для одного элемента данных

Можно определить несколько правил низкоуровневого обнаружения с одним и тем же элементом данных обнаружения.

Для этого необходимо определить параметр агента Alias, позволяющий использовать изменённые ключи элементов данных обнаружения в разных правилах обнаружения, например: `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` и т. д.

Ограничения на возвращаемые значения

Для JSON-данных правила обнаружения низкого уровня ограничений нет, если они получены напрямую сервером Zabbix. Это связано с тем, что возвращаемые значения обрабатываются без сохранения в базе данных.

Для пользовательских правил обнаружения низкого уровня ограничений также нет. Однако если данные пользовательского правила обнаружения низкого уровня получаются с помощью пользовательского параметра, применяется ограничение на возвращаемое значение пользовательского параметра.

Если данные должны проходить через прокси Zabbix, прокси должен сохранить эти данные в базе данных. В таком случае применяются ограничения базы данных.

Примеры правил обнаружения

В этом разделе приведены практические примеры настройки правил обнаружения для разных типов сущностей, таких как файловые системы, сетевые интерфейсы, службы, датчики, объекты SNMP, Java-приложения и базы данных.

Каждый пример показывает, как может быть структурировано правило обнаружения, включая ключ элемента данных или используемый запрос, JSON-вывод с макросами LLD и то, как эти результаты используются для создания прототипов элементов данных, триггеров и графиков.

- Обнаружение блочных устройств
- Обнаружение CPU и ядер CPU
- Обнаружение интерфейсов узла сети в Zabbix
- Обнаружение датчиков IPMI
- Обнаружение объектов JMX
- Обнаружение смонтированных файловых систем
- Обнаружение сетевых интерфейсов
- Обнаружение OID SNMP
- Обнаружение OID SNMP (устаревшее)
- Обнаружение служб systemd
- Обнаружение экземпляров счетчиков производительности Windows
- Обнаружение служб Windows
- Обнаружение с использованием запросов WMI
- Обнаружение с использованием SQL-запросов ODBC
- Обнаружение с использованием данных Prometheus

Пользовательские правила LLD

Обзор

Также можно создать полностью пользовательское правило LLD, обнаруживающее любой тип сущностей — например, базы данных на сервере баз данных.

Для этого следует создать пользовательский элемент данных, который возвращает строку JSON, указывая найденные объекты и, при необходимости, некоторые их свойства. Количество макросов на одну сущность не ограничено — в то время как встроенные правила обнаружения возвращают либо один, либо два макроса (например, два для обнаружения файловых систем), можно вернуть и больше.

Пример

Требуемый формат строки JSON лучше всего иллюстрируется на примере. Предположим, что вы используете старый агент Zabbix 1.8 (тот, который не поддерживает ключ `vfs.fs.discovery`), но вам все еще нужно обнаруживать файловые системы. Ниже приведен простой Perl-скрипт для Linux, который обнаруживает смонтированные файловые системы и выводит данные в JSON, включая имя файловой системы и ее тип. Один из способов использовать его — как UserParameter с ключом `vfs.fs.discovery_perl`:

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;
```

```
    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"{#FSNAME}\":\"$fsname\",\n";
    print "\t\t\"{#FSTYPE}\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Attention:
Допустимые символы для имен макросов LLD: **0-9** , **A-Z** , _ , **.**. Строчные буквы в именах не поддерживаются.

Ниже показан пример его вывода (переформатирован для наглядности). JSON для пользовательских проверок обнаружения должен соответствовать тому же формату.

```
[
    { "{#FSNAME}":"/",                          "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                       "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                      "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                       "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",                   "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",                   "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                      "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

В предыдущем примере требуется, чтобы ключи совпадали с именами макросов LLD, используемыми в прототипах; альтернативный вариант — извлекать значения макросов LLD с помощью JSONPath {#FSNAME} → $.fsname и {#FSTYPE} → $.fstype, что делает возможным такой скрипт:

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\":\"$fsname\",\n";
    print "\t\t\"fstype\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Ниже показан пример его вывода (переформатирован для наглядности).

```
[
    { "fsname":"/",                         "fstype":"rootfs"   },
    { "fsname":"/sys",                      "fstype":"sysfs"    },
    { "fsname":"/proc",                     "fstype":"proc"     },
    { "fsname":"/dev",                      "fstype":"devtmpfs" },
```

```
    { "fsname":"/dev/pts",                  "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",              "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                  "fstype":"tmpfs"    },
    { "fsname":"/home",                     "fstype":"ext3"     },
    { "fsname":"/tmp",                      "fstype":"ext3"     },
    { "fsname":"/usr",                      "fstype":"ext3"     },
    { "fsname":"/var",                      "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",  "fstype":"fusectl"  }
]
```

Затем в поле *Filter* правила обнаружения можно указать `{#FSTYPE}` как макрос и `rootfs|ext3` как регулярное выражение.

Note:
В пользовательских правилах LLD не обязательно использовать имена макросов вроде FSNAME/FSTYPE; вы можете использовать любые имена по своему усмотрению. Если используется JSONPath, то строка LLD будет элементом массива, который может быть объектом, но также может быть и другим массивом или значением.

Обратите внимание: при использовании пользовательского параметра возвращаемое значение ограничено 16 МБ. Подробнее см. ограничения на данные для возвращаемых значений LLD.

Мониторинг виртуальных машин

Обзор Zabbix может использовать правила низкоуровневого обнаружения для автоматического обнаружения гипервизоров и виртуальных машин VMware, а также создавать узлы сети для их мониторинга на основе предварительно определённых прототипов узлов сети.

Zabbix также включает готовые к использованию шаблоны для мониторинга гипервизоров VMware vCenter или ESXi.

Минимальная необходимая версия VMware vCenter или vSphere — 5.1.

Сбор данных Мониторинг виртуальных машин выполняется в два шага:

1. Процессы Zabbix *vmware collector* собирают данные виртуальных машин — процессы получают необходимую информацию от веб-сервисов VMware по протоколу SOAP, предварительно обрабатывают её и сохраняют в разделяемой памяти сервера Zabbix.
2. Процессы Zabbix *poller* извлекают данные при помощи ключей элементов данных мониторинга VMware простых проверок Zabbix.

Zabbix делит собранные данные на данные конфигурации VMware и данные счётчиков производительности VMware. Оба типа данных собираются независимо процессами *vmware collector*.

На основе данных счётчиков производительности VMware доступна статистика по следующим категориям:

- Хранилище данных
- Дисковое устройство
- Процессор
- Питание
- Сетевой интерфейс
- Элементы данных пользовательских счётчиков производительности

Для полного списка элементов данных, которые получают данные от счётчиков производительности VMware, обратитсесь к разделу *Ключи элементов данных мониторинга VMware*.

Обратите внимание, что частота получения событий VMware зависит от интервала опроса vmware.eventlog, но не может быть меньше 5 секунд.

Конфигурация Если сервер Zabbix собран из исходных кодов, его необходимо собрать с параметрами конфигурации `--with-libcurl --with-libxml2`, чтобы включить мониторинг виртуальных машин. Пакеты Zabbix уже собраны с включенными этими параметрами.

Для мониторинга виртуальных машин можно изменить следующие параметры файла конфигурации сервера Zabbix:

- StartVMwareCollectors

Note:
Рекомендуется включить больше сборщиков, чем количество отслеживаемых служб VMware; в противном случае получение статистики счетчиков производительности VMware может задерживаться из-за получения данных конфигурации VMware (что занимает некоторое время в крупных установках).

 Как правило, значение StartVMwareCollectors не должно быть меньше 2 и не должно превышать удвоенное количество отслеживаемых служб VMware: Количество служб < StartVMwareCollectors < (Количество служб * 2). Например, при мониторинге одной службы VMware задайте для StartVMwareCollectors значение 2; при мониторинге трех служб — 5.

 Обратите внимание, что требуемое количество сборщиков также зависит от масштаба среды VMware, а также от параметров конфигурации VMwareFrequency и VMwarePerfFrequency.

- VMwareCacheSize
- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
Чтобы поддерживать метрики емкости хранилищ данных, убедитесь, что значение ключа VMware vpxd.stats.maxQueryMetrics установлено не ниже 64. Дополнительные сведения см. в статье базы знаний VMware.

Обнаружение

Zabbix может использовать правила обнаружения низкого уровня (например, vmware.hv.discovery[{$VMWARE.URL}]) для автоматического обнаружения гипервизоров VMware и виртуальных машин. Кроме того, Zabbix может использовать прототипы узлов сети для автоматического создания реальных узлов сети для обнаруженных сущностей. Дополнительную информацию см. в *Прототипы узлов сети*.

Примеры конфигурации

Для базового примера настройки Zabbix для мониторинга VMware с использованием шаблона *VMware FQDN* см. *Мониторинг VMware с помощью Zabbix*.

Для более подробного примера создания узла сети, правила обнаружения низкого уровня и прототипа узла сети для мониторинга VMware см. *Пример настройки*.

Расширенное журналирование Данные, собираемые процессами *vmware collector*, можно записывать в журнал для подробной отладки с использованием уровня отладки 5.
Уровень отладки можно настроить в конфигурационных файлах сервера и прокси или с помощью параметра управления во время выполнения -R log_level_increase="vmware collector,N", где N — номер процесса.

Например, чтобы повысить уровень отладки с 4 до 5 для всех процессов *vmware collector*, выполните следующую команду:

```
zabbix_server -R log_level_increase="vmware collector"
```

Чтобы повысить уровень отладки с 4 до 5 для второго процесса *vmware collector*, выполните следующую команду:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Когда расширенное журналирование данных VMware больше не требуется, рекомендуется вернуть уровень отладки к значению по умолчанию (3), выполнив команду -R log_level_decrease.

Устранение неполадок

- В случае недоступности метрик убедитесь, что они не стали недоступны или не были отключены по умолчанию в последних версиях VMware vSphere, а также что на запросы к базе данных метрик производительности не наложены какие-либо ограничения. Дополнительные сведения см. в ZBX-12094.
- Если config.vpxd.stats.maxQueryMetrics недопустим или возникает ошибка превышения максимального допустимого числа символов, добавьте параметр config.vpxd.stats.maxQueryMetrics в настройки vCenter Server. Значение этого параметра должно быть таким же, как значение maxQuerysize в файле web.xml VMware. Дополнительные сведения см. в статье *VMware Knowledge Base*.
- Если вы подозреваете, что ваша установка Zabbix использует слишком много памяти, см. Профилирование чрезмерного использования памяти с tcmalloc.

Ключи элементов данных для мониторинга VMware

Обзор На этой странице представлены подробные сведения о простых проверках, которые можно использовать для мониторинга сред виртуализации VMware.

Метрики сгруппированы по целям мониторинга.

Поддерживаемые ключи элементов данных Ключи элементов данных перечислены без параметров и дополнительной информации. Щелкните ключ элемента данных, чтобы увидеть полные сведения.

Ключ элемента данных	Описание	Группа элементов данных
vmware.eventlog	Журнал событий VMware.	Общие службы
vmware.fullname	Полное имя службы VMware.	
vmware.version	Версия службы VMware.	
vmware.cl.perfcounter	Метрики счетчика производительности кластера VMware.	Cluster
vmware.cluster.alarms.get	Данные тревог кластера VMware.	
vmware.cluster.discovery	Обнаружение кластеров VMware.	
vmware.cluster.property	Свойство кластера VMware.	
vmware.cluster.status	Состояние кластера VMware.	
vmware.cluster.tags.get	Массив тегов кластера VMware.	
vmware.datastore.alarms.get	Данные тревог хранилища данных VMware.	Хранилище данных
vmware.datastore.discovery	Обнаружение хранилищ данных VMware.	
vmware.datastore.hv.list	Список гипервизоров хранилища данных.	
vmware.datastore.perfcounter	Значение счетчика производительности хранилища данных VMware.	
vmware.datastore.property	Свойство хранилища данных VMware.	
vmware.datastore.read	Время операции чтения из хранилища данных.	
vmware.datastore.size	Объем хранилища данных VMware в байтах или в процентах от общего объема.	
vmware.datastore.tags.get	Массив тегов хранилища данных VMware.	
vmware.datastore.write	Время операции записи в хранилище данных.	
vmware.dc.alarms.get	Данные тревог центра обработки данных VMware.	Центр обработки данных
vmware.dc.discovery	Обнаружение центров обработки данных VMware.	
vmware.dc.tags.get	Массив тегов центра обработки данных VMware.	
vmware.dvswitch.discovery	Обнаружение распределенных коммутаторов VMware vSphere Distributed Switch.	vSphere Distributed Switch
vmware.dvswitch.fetchportsget	Данные портов VMware vSphere Distributed Switch.	
vmware.hv.alarms.get	Данные тревог гипервизора VMware.	Гипервизор
vmware.hv.cluster.name	Имя кластера гипервизора VMware.	
vmware.hv.connectionstate	Состояние подключения гипервизора VMware.	
vmware.hv.cpu.usage	Использование процессора гипервизора VMware (Hz).	
vmware.hv.cpu.usage.perf	Использование процессора гипервизора VMware в процентах за интервал.	
vmware.hv.cpu.utilization	Использование процессора гипервизора VMware в процентах за интервал, зависит от управления питанием или HT.	
vmware.hv.datacenter.name	Имя центра обработки данных гипервизора VMware.	
vmware.hv.datastore.discovery	Обнаружение хранилищ данных гипервизора VMware.	
vmware.hv.datastore.list	Список хранилищ данных гипервизора VMware.	
vmware.hv.datastore.multipath	Количество доступных путей к хранилищу данных.	
vmware.hv.datastore.read	Среднее время операции чтения из хранилища данных.	
vmware.hv.datastore.size	Объем хранилища данных VMware в байтах или в процентах от общего объема.	
vmware.hv.datastore.write	Среднее время операции записи в хранилище данных.	
vmware.hv.discovery	Обнаружение гипервизоров VMware.	
vmware.hv.diskinfo.get	Данные о дисках гипервизора VMware.	
vmware.hv.fullname	Полное имя продукта, включая сведения о версии.	
vmware.hv.hw.cpu.freq	Частота процессора гипервизора VMware.	
vmware.hv.hw.cpu.model	Модель процессора гипервизора VMware.	
vmware.hv.hw.cpu.num	Количество ядер процессора на гипервизоре VMware.	
vmware.hv.hw.cpu.threads	Количество потоков процессора на гипервизоре VMware.	

Ключ элемента данных	Описание	Группа элементов данных
vmware.hv.hw.memory	Общий объем памяти гипервизора VMware.	
vmware.hv.hw.model	Модель гипервизора VMware.	
vmware.hv.hw.sensors.get	Значение аппаратных датчиков гипервизора VMware.	
vmware.hv.hw.serialnumber	Серийный номер гипервизора VMware.	
vmware.hv.hw.uuid	UUID BIOS гипервизора VMware.	
vmware.hv.hw.vendor	Имя производителя гипервизора VMware.	
vmware.hv.maintenance	Состояние обслуживания гипервизора VMware.	
vmware.hv.memory.size.ballooned	Объем balloon-памяти гипервизора VMware.	
vmware.hv.memory.used	Используемый объем памяти гипервизора VMware.	
vmware.hv.net.if.discovery	Обнаружение сетевых интерфейсов гипервизора VMware.	
vmware.hv.network.in	Статистика входящего сетевого трафика гипервизора VMware.	
vmware.hv.network.linkspeed	Скорость сетевого интерфейса гипервизора VMware.	
vmware.hv.network.out	Статистика исходящего сетевого трафика гипервизора VMware.	
vmware.hv.perfcounter	Значение счетчика производительности гипервизора VMware.	
vmware.hv.property	Свойство гипервизора VMware.	
vmware.hv.power	Потребление энергии гипервизором VMware.	
vmware.hv.sensor.health.status	Сводное состояние здоровья датчика гипервизора VMware.	
vmware.hv.sensors.get	Датчики состояния аппаратного обеспечения производителя гипервизора VMware.	
vmware.hv.status	Состояние гипервизора VMware.	
vmware.hv.tags.get	Массив тегов гипервизора VMware.	
vmware.hv.uptime	Время непрерывной работы гипервизора VMware.	
vmware.hv.version	Версия гипервизора VMware.	
vmware.hv.vm.num	Количество виртуальных машин на гипервизоре VMware.	
vmware.rp.cpu.usage	Использование CPU в герцах за интервал в VMware Resource Pool.	Пул ресурсов
vmware.rp.memory	Показатели памяти пула ресурсов VMware.	
vmware.alarms.get	Данные тревог виртуального центра VMware.	Виртуальный центр
vmware.vm.alarms.get	Данные тревог виртуальной машины VMware.	Виртуальная машина
vmware.vm.attribute	Значение пользовательского атрибута виртуальной машины VMware.	
vmware.vm.cluster.name	Имя виртуальной машины VMware.	
vmware.vm.consolidationneeded	Диск виртуальной машины VMware требует консолидации.	
vmware.vm.cpu.latency	Процент времени, в течение которого виртуальная машина не может выполняться, поскольку конкурирует за доступ к физическим CPU.	
vmware.vm.cpu.num	Количество процессоров на виртуальной машине VMware.	
vmware.vm.cpu.readiness	Процент времени, когда виртуальная машина была готова, но не могла быть запланирована для выполнения на физическом CPU.	
vmware.vm.cpu.ready	Время, когда виртуальная машина была готова, но не могла быть запланирована для выполнения на физическом CPU.	
vmware.vm.cpu.swapwait	Процент времени CPU, затраченного на ожидание подкачки из swap.	
vmware.vm.cpu.usage	Использование процессора виртуальной машины VMware (Hz).	
vmware.vm.cpu.usage.perf	Использование процессора виртуальной машины VMware в процентах за интервал.	
vmware.vm.datacenter.name	Имя центра обработки данных виртуальной машины VMware.	
vmware.vm.discovery	Обнаружение виртуальных машин VMware.	
vmware.vm.guest.memory.size.swapped	Объем физической памяти гостевой ОС, выгруженной в пространство подкачки.	
vmware.vm.guest.osuptime	Общее время, прошедшее с момента последней загрузки операционной системы.	
vmware.vm.hv.maintenance	Состояние обслуживания гипервизора виртуальной машины VMware.	
vmware.vm.hv.name	Имя гипервизора виртуальной машины VMware.	
vmware.vm.memory.size	Общий объем памяти виртуальной машины VMware.	
vmware.vm.memory.size.ballooned	Объем balloon-памяти виртуальной машины VMware.	
vmware.vm.memory.size.compressed	Объем сжатой памяти виртуальной машины VMware.	
vmware.vm.memory.size.consumed	Объем физической памяти узла сети, используемой для хранения страниц физической памяти гостевой ОС.	
vmware.vm.memory.size.private	Объем частной памяти виртуальной машины VMware.	
vmware.vm.memory.size.shared	Объем общей памяти виртуальной машины VMware.	

Ключ элемента данных	Описание	Группа элементов данных
vmware.vm.memory.size.swapped	Объем памяти виртуальной машины VMware, выгруженной в swap.	
vmware.vm.memory.size.usage.guest	Использование памяти гостевой ОС виртуальной машины VMware.	
vmware.vm.memory.size.usage.host	Использование памяти узла сети виртуальной машины VMware.	
vmware.vm.memory.usage	Процент использованной физической памяти узла сети.	
vmware.vm.net.if.discovery	Обнаружение сетевых интерфейсов виртуальной машины VMware.	
vmware.vm.net.if.in	Статистика входящего трафика сетевого интерфейса виртуальной машины VMware.	
vmware.vm.net.if.out	Статистика исходящего трафика сетевого интерфейса виртуальной машины VMware.	
vmware.vm.net.if.usage	Использование сети виртуальной машины VMware за интервал.	
vmware.vm.perfcounter	Значение счетчика производительности виртуальной машины VMware.	
vmware.vm.powerstate	Состояние питания виртуальной машины VMware.	
vmware.vm.property	Свойство виртуальной машины VMware.	
vmware.vm.snapshot.get	Состояние снимка виртуальной машины VMware.	
vmware.vm.state	Состояние виртуальной машины VMware.	
vmware.vm.storage.committed	Зафиксированный объем хранилища виртуальной машины VMware.	
vmware.vm.storage.readoio	Среднее количество ожидающих запросов на чтение к виртуальному диску за интервал сбора.	
vmware.vm.storage.totalreadlatency	Среднее время чтения с виртуального диска.	
vmware.vm.storage.totalwritelatency	Среднее время записи на виртуальный диск.	
vmware.vm.storage.uncommitted	Незафиксированный объем хранилища виртуальной машины VMware.	
vmware.vm.storage.unshared	Неразделяемый объем хранилища виртуальной машины VMware.	
vmware.vm.storage.writeoio	Среднее количество ожидающих запросов на запись к виртуальному диску за интервал сбора.	
vmware.vm.tags.get	Массив тегов виртуальной машины VMware.	
vmware.vm.tools	Состояние или версия guest tools виртуальной машины VMware.	
vmware.vm.uptime	Время непрерывной работы виртуальной машины VMware.	
vmware.vm.vfs.dev.discovery	Обнаружение дисковых устройств виртуальной машины VMware.	
vmware.vm.vfs.dev.read	Статистика чтения дискового устройства виртуальной машины VMware.	
vmware.vm.vfs.dev.write	Статистика записи дискового устройства виртуальной машины VMware.	
vmware.vm.vfs.fs.discovery	Обнаружение файловых систем виртуальной машины VMware.	
vmware.vm.vfs.fs.size	Статистика файловой системы виртуальной машины VMware.	

Подробности ключей элементов данных Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками **< >**, опциональны.

vmware.eventlog[url,<mode>,<severity>]

 Журнал событий VMware.
 Возвращаемое значение: *Log*.

Параметры:

- **url** - URL службы VMware;
- **mode** - *all* (по умолчанию) или *skip* - пропуск обработки более старых данных;
- **severity** - фильтрация по уровню серьезности: *error*, *warning*, *info* или *user*. Этот параметр должен быть заключен в кавычки, если указано более одного уровня серьезности в списке, разделенном запятыми (например,`"error,warning,info,u`
  По умолчанию отключено.

Комментарии:

- Для каждого URL может быть только один ключ элемента данных `vmware.eventlog`;
- См. также пример фильтрации записей журнала событий VMware.

vmware.fullnameurl

 Полное имя службы VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware.

vmware.versionurl

 Версия службы VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware.

vmware.cl.perfcounter[url,id,путь,<экземпляр>]

 Метрики счётчиков производительности VMware кластера.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **id** — ID кластера VMware. `id` может быть получено из `vmware.cluster.discovery[]` как {#CLUSTER.ID}.
- **путь** — путь к счётчику производительности[1];
- **экземпляр** — экземпляр счётчика производительности.

vmware.cluster.alarms.get[url,id]

 Данные alarm'ов кластера VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **id** — ID кластера VMware.

vmware.cluster.discoveryurl

 Обнаружение кластеров VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware.

vmware.cluster.property[url,id,prop]

 Свойство кластера VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** - URL службы VMware;
- **id** - ID кластера VMware;
- **prop** - путь к свойству, которое является именем свойства объекта VM, как определено в VMware SDK.

Примеры:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,имя]

 Статус кластера VMware.
 Возвращаемое значение: *0* — серый; *1* — зелёный; *2* — жёлтый; *3* — красный.

Параметры:

- **url** — URL службы VMware;
- **имя** — имя кластера VMware.

vmware.cluster.tags.get[url,id]

 Массив тегов кластера VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **id** — ID кластера VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.datastore.alarms.get[url,uuid]

 Данные alarm'ов хранилищ данных VMware.
 Возвращаемое значение: Объект JSON.

Parameters:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware.

vmware.datastore.discovery[url,<filter_uuid>]

 Обнаружение хранилищ данных VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **filter_uuid** — один уникальный идентификатор HV или VM (по умолчанию: пусто). Если указано, будут обнаружены только присоединённые хранилища данных.

vmware.datastore.hv.list[url,хранилище]

 Список гипервизоров по хранилищу.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **хранилище** — UUID или имя хранилища.

Пример вывода:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,путь,<экземпляр>]

 Значение счётчика производительности хранилища данных VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — уникальный глобальный идентификатор хранилища данных VMware;
- **путь** — путь к счётчику производительности[1];
- **экземпляр** — экземпляр счётчика производительности. Оставьте пустым для получения агрегированных значений (по умолчанию). может быть получен из `vmware.datastore.discovery[]` как часть массива {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 Свойство хранилища данных VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware;
- **prop** — путь свойства.

vmware.datastore.read[url,хранилище,<режим>]

 Время операций чтения из хранилища (в миллисекундах).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — *latency* (среднее значение, по умолчанию) или *maxlatency* (максимальное значение).

vmware.datastore.size[url,хранилище,<режим>]

 Размер хранилища данных VMware в байтах или в процентах от общего размера.
 Возвращаемое значение: *Целое число* — для байт; *Дробное число* — для процентов.

Параметры:

- **url** — URL службы VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — возможные значения: *total* (по умолчанию), *free*, *pfree* (свободно, в процентах), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 Массив тегов хранилища данных VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.datastore.write[url,хранилище,<режим>]

 Время операций записи в хранилище (в миллисекундах).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — *latency* (среднее значение, по умолчанию) или *maxlatency* (максимальное значение).

vmware.dc.alarms.get[url,id]

 Данные alarm'ов датацентра VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **id** — ID датацентра VMware.

vmware.dc.discoveryurl

 Обнаружение датацентров VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware.

vmware.dc.tags.get[url,id]

 Массив тегов датацентра VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **id** — ID датацентра VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.dvswitch.discoveryurl

 Обнаружение распределённых свитчей VMware vSphere.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware.

vmware.dvswitch.fetchports.get[url,uuid,<фильтр>,<режим>]

 Данные портов распределённого свитча VMware vSphere.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор распределённого свитча VMware.
- **фильтр** — одна строка с критериями выбора портов, перечисленными через запятую;
- **режим** — *state* (все XML без узлов XML "config", по умолчанию) или *full*.

Параметр **фильтр** поддерживает критерии [en], доступные в объекте данных VMware DistributedVirtualSwitchPortCriteria.

Пример:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 Данные alarm'ов гипервизора VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cluster.name[url,uuid]

 Имя кластера гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.connectionstate[url,uuid]

 Состояние соединения гипервизора VMware.
 Возвращаемое значение: *Строка*: *connected*, *disconnected* или *notResponding*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.usage[url,uuid]

 Использование процессора гипервизором VMware (Гц).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.usage.perf[url,uuid]

 Использование процессора гипервизора VMware в процентном отношении за интервал времени.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.utilization[url,uuid]

 Использование процессора гипервизора VMware в процентном отношении за интервал времени, зависит от управления питанием или HT.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datacenter.name[url,uuid]

 Имя датацентра гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datastore.discovery[url,uuid]

 Обнаружение хранилищ у гипервизора VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datastore.list[url,uuid]

 Список хранилищ гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Пример вывода:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<хранилище>,<partitionid>]

 Количество доступных путей до хранилища.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;

- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **хранилище** — UUID или имя хранилища;
- **partitionid** — внутренний ID физического устройства из `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,хранилище,<режим>]

 Среднее время операций чтения из хранилища данных (миллисекунд).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — *latency* (по умолчанию).

vmware.hv.datastore.size[url,uuid,хранилище,<режим>]

 Размер пространства VMware хранилища в байтах или в процентах от общего размера.
 Возвращаемое значение: *Целое число* — для байт; *Число с плавающей точкой* — для процентов.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — возможные значения: *total* (по умолчанию), *free*, *pfree* (свободно, в процентах), *uncommitted*.

vmware.hv.datastore.write[url,uuid,хранилище,<режим>]

 Среднее время операций записи в хранилище данных (миллисекунд).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **хранилище** — UUID или имя хранилища;
- **режим** — *latency* (по умолчанию).

vmware.hv.discoveryurl

 Обнаружение гипервизоров VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware.

vmware.hv.diskinfo.get[url,uuid]

 Данные о дисках гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.fullname[url,uuid]

 Полное наименование продукта, включая информацию о версии.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.freq[url,uuid]

 Частота процессора гипервизора VMware (Гц).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.model[url,uuid]

 Модель процессора гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.num[url,uuid]

 Количество ядер процессора у гипервизора VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.threads[url,uuid]

 Количество потоков процессора у гипервизора VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.memory[url,uuid]

 Общее количество памяти гипервизора VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.model[url,uuid]

 Модель гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.sensors.get[url,uuid]

 Значения датчиков оборудования гипервизора VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.serialnumber[url,uuid]

 Серийный номер гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere API 6.7 и новее.

vmware.hv.hw.uuid[url,uuid]

 UUID BIOS гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere API 6.7 и новее.

vmware.hv.hw.vendor[url,uuid]

 Имя поставщика гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere API 6.7 и новее.

vmware.hv.maintenance[url,uuid]

 Состояние обслуживания гипервизора VMware.
 Возвращаемое значение: *0* — не в обслуживании; *1* — в обслуживании.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.memory.size.ballooned[url,uuid]

 Размер раздутой памяти гипервизора VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.memory.used[url,uuid]

 Размер используемой гипервизором VMware памяти (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url**— URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.net.if.discovery[url,uuid]

 Обнаружение сетевых интерфейсов гипервизора VMware.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.network.in[url,uuid,<режим>]

 Статистика по входящему трафику на сетевом интерфейсе гипервизора VMware (в байтах в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **режим** — *bps* (по умолчанию), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 Скорость сетевого интерфейса гипервизора VMware.
 Возвращаемое значение: *Целое число*. Если сетевой интерфейс не поднят, возвращает *0*, в противном случае — значение скорости интерфейса.

Parameters:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **ifname** — имя интерфейса.

vmware.hv.network.out[url,uuid,<режим>]

 Статистика по исходящему трафику на сетевом интерфейсе гипервизора VMware (в байтах в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **режим** — *bps* (по умолчанию), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,путь,<экземпляр>]

 Значение счётчика производительности гипервизора VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;

- **uuid** — уникальный глобальный идентификатор хранилища данных VMware;
- **путь** — путь к счётчику производительности[1];
- **экземпляр** — экземпляр счётчика производительности. Оставьте пустым для получения агрегированных значений (по умолчанию).

vmware.hv.property[url,uuid,prop]

 Свойство гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **prop** — путь свойства.

vmware.hv.power[url,uuid,<max>]

 Энергопотребление гипервизора VMware (Вт).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **max** — максимально допустимое энергопотребление .

vmware.hv.sensor.health.state[url,uuid]

 Сводный датчик состояния работоспособности гипервизора VMware.
 Возвращаемое значение: *Целое число*: *0* — серый; *1* — зелёный; *2* — жёлтый; *3* — красный.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Обратите внимание, что этот элемент данных может не работать в VMware vSphere 6.5 и более поздних версиях, поскольку VMware прекратила поддержку датчика *VMware Rollup Health State*.

vmware.hv.sensors.get[url,uuid]

 Датчики состояний поставщика оборудования гипервизора VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.status[url,uuid]

 Состояние гипервизора VMware.
 Возвращаемое значение: *Целое число*: *0* — серый; *1* — зелёный; *2* — жёлтый; *3* — красный.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных использует свойство общего статуса хост-системы.

vmware.hv.tags.get[url,uuid]

 Массив тегов гипервизора VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.hv.uptime[url,uuid,<max>]

 Время работы гипервизора VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных использует свойство общего статуса хост-системы.

vmware.hv.version[url,uuid]

 Версия гипервизора VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.vm.num[url,uuid]

 Количество виртуальных машин на гипервизоре VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.rp.cpu.usage[url,rpid]

 Использование CPU (в герцах) за интервал в пуле ресурсов VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **rpid** — ID пула ресурсов VMware.

vmware.rp.memory[url,rpid,<режим>]

 Метрики памяти пула ресурсов VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **rpid** — ID пула ресурсов VMware;
- **режим** — возможные значения:
consumed (по умолчанию) — объем физической памяти хоста, потребляемый для поддержания страниц гостевой физической памяти
 ballooned — объем гостевой физической памяти, восстановленной из виртуальной машины драйвером balloon в гостевой системе
overhead — физическая память хоста, потребляемая структурами данных ESXi для работы виртуальных машин

vmware.alarms.geturl

 Данные alarm’ов виртуального центра VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware

vmware.vm.alarms.get[url,uuid]

 Данные alarm’ов виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.attribute[url,uuid,name]

 Значение пользовательского атрибута виртуальной машины VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware;
- **name** — имя пользовательского атрибута.

vmware.vm.cluster.name[url,uuid]

 Имя виртуальной машины VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.consolidationneeded[url,uuid]

 Диск виртуальной машины VMware требует консолидации.
 Возвращаемое значение: *Строка*: *true* — требуется консолидация; *false* — консолидация не требуется.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware

vmware.vm.cpu.latency[url,uuid]

 Процент времени, в течение которого виртуальная машина не может работать из-за того, что борется за доступ к физическому(-им) CPU.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.num[url,uuid]

 Количество процессоров на виртуальной машине VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.readiness[url,uuid,<экземпляр>]

 Процент времени, в течение которого виртуальная машина была готова, но не получилось ее запланировать на запуск на физическом CPU.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр CPU.

vmware.vm.cpu.ready[url,uuid]

 Время (в миллисекундах), когда виртуальная машина была готова, но не запланирована на выполнение на физическом CPU. Это время зависит от количества виртуальных машин на хосте и загрузки их CPU (%).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.swapwait[url,uuid,<экземпляр>]

 Процент времени CPU, потраченного на ожидание замены (swap-in).
 Возвращаемое значение: *Дробное число*.

Parameters:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр CPU.

vmware.vm.cpu.usage[url,uuid]

 Использование процессора виртуальной машины VMware (Гц).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.usage.perf[url,uuid]

 Использование процессора виртуальной машины VMware в процентах в течение интервала.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;

- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.datacenter.name[url,uuid]

 Имя центра обработки данных виртуальной машины VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.discoveryurl

 Обнаружение виртуальных машин VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware.

vmware.vm.guest.memory.size.swapped[url,uuid]

 Объём гостевой физической памяти, которая выгружается в файл подкачки (КБ).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.guest.osuptime[url,uuid]

 Общее количество времени с момента последней загрузки операционной системы (в секундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.hv.maintenance[url,uuid]

 Статус обслуживания гипервизора виртуальной машины VMware.
 Возвращаемое значение: *0* — не в режиме обслуживания; *1* — в режиме обслуживания.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.hv.name[url,uuid]

 Имя гипервизора виртуальной машины VMware.
 Возвращаемое значение: *Строка*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size[url,uuid]

 Общий размер памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.ballooned[url,uuid]

 Размер раздутой памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.compressed[url,uuid]

 Размер сжатой памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.consumed[url,uuid]

 Объём физической памяти гипервизора (хоста), потребляемый для поддержания страниц физической памяти гостевой ОС (КБ).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.private[url,uuid]

 Размер приватной памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.shared[url,uuid]

 Размер разделяемой памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.swapped[url,uuid]

 Размер памяти, выгруженной в файл подкачки виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 Размер памяти, используемой виртуальной машиной VMware (статистика гостевой виртуальной машины, в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.usage.host[url,uuid]

 Размер памяти, используемой виртуальной машиной VMware (статистика гипервизора, в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.usage[url,uuid]

 Процент потреблённой физической памяти гипервизора.
 Возвращаемое значение: *Дробное число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.net.if.discovery[url,uuid]

 Обнаружение сетевых интерфейсов виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;

- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.net.if.in[url,uuid,экземпляр,<режим>]

 Статистика по входящему трафику на сетевом интерфейсе виртуальной машины VMware (в байтах/пакетах в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр сетевого интерфейса;
- **режим** — *bps* (по умолчанию) или *pps* — байт либо пакетов в секунду.

vmware.vm.net.if.out[url,uuid,экземпляр,<режим>]

 Статистика по исходящему трафику на сетевом интерфейсе виртуальной машины VMware (в байтах/пакетах в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр сетевого интерфейса;
- **режим** — *bps* (по умолчанию) или *pps* — байт либо пакетов в секунду.

vmware.vm.net.if.usage[url,uuid,<экземпляр>]

 Утилизация сети виртуальной машиной VMware (суммарная статистика по входящему и исходящему трафику) в течение интервала (КБ/с).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр сетевого интерфейса.

vmware.vm.perfcounter[url,uuid,путь,<экземпляр>]

 Значение счётчика производительности у виртуальной машины VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — уникальный глобальный идентификатор виртуальной машины VMware;
- **путь** — путь к счётчику производительности[1];
- **экземпляр** — экземпляр счётчика производительности. Оставьте пустым для получения агрегированных значений (по умолчанию).

vmware.vm.powerstate[url,uuid]

 Состояние питания виртуальной машины VMware.
 Возвращаемое значение: *0* — выключено (poweredOff); *1* — включено (poweredOn); *2* — приостановлено (suspended).

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.property[url,uuid,prop]

 Свойство виртуальной машины VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** - URL службы VMware;
- **uuid** - глобальный уникальный идентификатор виртуальной машины VMware;
- **prop** - путь свойства.

vmware.vm.snapshot.get[url,uuid]

 Состояние снапшота виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;

- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.state[url,uuid]

 Состояние виртуальной машины VMware.
 Возвращаемое значение: *Строка*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* или *unknown*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.committed[url,uuid]

 Объём записанных данных виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.readoio[url,uuid,экземпляр]

 Среднее количество невыполненных запросов на чтение к виртуальному диску в течение интервала сбора данных.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Среднее время чтения с виртуального диска (в миллисекундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Среднее время записи на виртуальный диск (в миллисекундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства.

vmware.vm.storage.uncommitted[url,uuid]

 Объём незаписанных данных виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.unshared[url,uuid]

 Размер неразделяемого пространства хранилища виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.writeoio[url,uuid,экземпляр]

 Среднее количество невыполненных запросов на запись на виртуальный диск в течение интервала сбора данных.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;

- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства.

vmware.vm.tags.get[url,uuid]

 Массив тегов виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.vm.tools[url,uuid,<режим>]

 Состояние гостевых инструментов виртуальной машины VMware.
 Возвращаемое значение: *Строка*.
 В режиме *status*: *guestToolsExecutingScripts* — VMware Tools запускаются; *guestToolsNotRunning* — VMware Tools не работают; *guestToolsRunning* — VMware Tools работают.
 В режиме *version*: версия.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **режим** — *version*, *status* (по умолчанию).

vmware.vm.uptime[url,uuid]

 Время непрерывной работы виртуальной машины VMware (в секундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.vfs.dev.discovery[url,uuid]

 Обнаружение дисковых устройств виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.vfs.dev.read[url,uuid,экземпляр,<режим>]

 Статистика чтения с дискового устройства виртуальной машины VMware (в байтах/операциях в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства;
- **режим** — *bps* (по умолчанию) или *ops* — байт или операций в секунду.

vmware.vm.vfs.dev.write[url,uuid,экземпляр,<режим>]

 Статистика записи на дисковое устройство виртуальной машины VMware (в байтах/операциях в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **экземпляр** — экземпляр дискового устройства;
- **режим** — *bps* (по умолчанию) или *ops* — байт или операций в секунду.

vmware.vm.vfs.fs.discovery[url,uuid]

 Обнаружение файловых систем виртуальной машины VMware.
 Возвращаемое значение: Объект JSON.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

Для работы этого элемента данных необходимо, чтобы на гостевой виртуальной машине были установлены VMware Tools.

vmware.vm.vfs.fs.size[url,uuid,fsname,<режим>]

 Статистика файловой системы виртуальной машины VMware (в байтах/процентах).
 Возвращаемое значение: *Целое число* — для байт; *Дробное число* — для процентов.

Параметры:

- **url** — URL службы VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **fsname** — имя файловой системы;
- **режим** — *total*, *free*, *used*, *pfree* или *pused*.

Для работы этого элемента данных необходимо, чтобы на гостевой виртуальной машине были установлены VMware Tools.

Сноски

[1] См. Создание пользовательских имен счетчиков производительности для VMware.

[2] Значение этих элементов данных получается из счетчиков производительности VMware, а параметр VMwarePerfFrequency используется для обновления их данных в кэше VMware Zabbix:

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in
- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

Дополнительная информация

Смотрите раздел мониторинга виртуальных машин для получения более подробной информации о настройке Zabbix для мониторинга сред VMware.

Поля ключей обнаружения виртуальных машин

В следующей таблице перечислены поля, возвращаемые ключами обнаружения, связанными с виртуальными машинами.

Ключ элемента данных		
Описание	**Поле**	**Получаемое содержимое**
vmware.cluster.discovery		
Выполняет обнаружение кластера.	{#CLUSTER.ID}	Идентификатор кластера.
	{#CLUSTER.NAME}	Имя кластера.

	”resource_pool”	Массив, содержащий данные resource pool, включая идентификатор группы ресурсов, массив тегов, путь resource pool, количество виртуальных машин.
		
		Структура массива:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Структуру массива `"tags"` см. в поле ”tags”.

Ключ элемента данных		
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.datastore.discovery		
Выполняет обнаружение хранилища данных.	{#DATASTORE}	Имя хранилища данных.
	{#DATASTORE.EXTENT}	Массив, содержащий идентификатор раздела extent хранилища данных и имя экземпляра. Структура массива: [{ "partitionid":1, "instance":"name" }]
	{#DATASTORE.TYPE}	Тип хранилища данных. Примеры значений: VMFS, NFS, vsan и т. д.
	{#DATASTORE.UUID}	Идентификатор хранилища данных.
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dc.discovery		
Выполняет обнаружение центра обработки данных.	{#DATACENTER}	Имя центра обработки данных.
	{#DATACENTERID}	Идентификатор центра обработки данных.
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dvswitch.discovery		
Выполняет обнаружение распределенных коммутаторов vSphere.	{#DVS.NAME}	Имя коммутатора.
	{#DVS.UUID}	Идентификатор коммутатора.
vmware.hv.discovery		
Выполняет обнаружение гипервизора.	{#HV.UUID}	Уникальный идентификатор гипервизора.
	{#HV.ID}	Идентификатор гипервизора (имя объекта управления HostSystem).
	{#HV.NAME}	Имя гипервизора.
	{#HV.NETNAME}	Сетевое имя хоста гипервизора.

Ключ элемента данных		
	{#HV.IP}	IP-адрес гипервизора, может быть пустым. В случае конфигурации HA с несколькими сетевыми интерфейсами соблюдается следующий приоритет выбора интерфейса: - предпочтение отдается IP, который находится в той же IP-подсети, что и IP vCenter; - предпочтение отдается IP из IP-подсети с шлюзом по умолчанию; - предпочтение отдается IP с интерфейса с наименьшим ID.
	{#CLUSTER.NAME}	Имя кластера, может быть пустым.
	{#DATACENTER.NAME}	Имя центра обработки данных.
	{#PARENT.NAME}	Имя контейнера, в котором хранится гипервизор.
	{#PARENT.TYPE}	Тип контейнера, в котором хранится гипервизор. Значения могут быть `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, где ”VMware” означает неизвестный тип контейнера.

	”resource_pool”	Массив, содержащий данные resource pool, включая идентификатор группы ресурсов, массив тегов, путь resource pool, количество виртуальных машин.
		
		Структура массива:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Структуру массива `"tags"` см. в поле ”tags”.

	”tags”	Массив, содержащий теги с именем тега, описанием и категорией.
		
		Структура массива:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.datastore.discovery		
Выполняет обнаружение хранилищ данных гипервизора. Обратите внимание, что несколько гипервизоров могут использовать одно и то же хранилище данных.	{#DATASTORE}	Имя хранилища данных.
	{#DATASTORE.TYPE}	Тип хранилища данных. Примеры значений: `VMFS`, `NFS`, `vsan` и т. д.
	{#DATASTORE.UUID}	Идентификатор хранилища данных.
	{#MULTIPATH.COUNT}	Зарегистрированное количество путей к хранилищу данных.
	{#MULTIPATH.PARTITION.COUNT}	Количество доступных дисковых разделов.

	”datastore_extent”	Массив, содержащий имя экземпляра extent хранилища данных и идентификатор раздела.
		
		Структура массива:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

Ключ элемента данных		
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.hv.net.if.discovery		
Выполняет обнаружение сетевых интерфейсов гипервизора.	{#IFNAME}	Имя интерфейса.
	{#IFDRIVER}	Драйвер интерфейса.
	{#IFDUPLEX}	Параметры дуплекса интерфейса.
	{#IFSPEED}	Скорость интерфейса.
	{#IFMAC}	MAC-адрес интерфейса.
vmware.vm.discovery		
Выполняет обнаружение виртуальных машин.	{#VM.UUID}	Уникальный идентификатор виртуальной машины.
	{#VM.ID}	Идентификатор виртуальной машины (имя объекта управления VirtualMachine).
	{#VM.NAME}	Имя виртуальной машины.
	{#HV.NAME}	Имя гипервизора.
	{#HV.UUID}	Уникальный идентификатор гипервизора.
	{#HV.ID}	Идентификатор гипервизора (имя объекта управления HostSystem).
	{#CLUSTER.NAME}	Имя кластера, может быть пустым.
	{#DATACENTER.NAME}	Имя центра обработки данных.
	{#DATASTORE.NAME}	Имя хранилища данных.
	{#DATASTORE.UUID}	Идентификатор хранилища данных.
	{#VM.IP}	IP-адрес виртуальной машины, может быть пустым.
	{#VM.DNS}	DNS-имя виртуальной машины, может быть пустым.
	{#VM.GUESTFAMILY}	Семейство гостевой ОС виртуальной машины, может быть пустым.
	{#VM.GUESTFULLNAME}	Полное имя гостевой ОС виртуальной машины, может быть пустым.
	{#VM.FOLDER}	Цепочка родительских папок виртуальной машины, может использоваться как значение для вложенных групп; имена папок объединяются с помощью ”/”. Может быть пустым.
	{#VM.TOOLS.STATUS}	Состояние VMware Tools виртуальной машины.
	{#VM.POWERSTATE}	Состояние питания виртуальной машины VMware (poweredOff, poweredOn или suspended).
	{#VM.RPOOL.ID}	Идентификатор resource pool.
	{#VM.RPOOL.PATH}	Полный путь resource pool без корневого имени ”Resources”. Имена папок объединяются с помощью ”/”.
	{#VM.SNAPSHOT.COUNT}	Количество снимков виртуальной машины.
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]

Ключ элемента данных		
	”vm_customattribute”	Массив пользовательских атрибутов виртуальной машины (если определены). Структура массива: [{ "name":"custom field name", "value":"custom field value" }]
	”net_if”	Массив сетевых интерфейсов виртуальной машины. Структура массива: [{ "ifname": "interface name", "ifdesc": "interface description", "ifmac": "00:00:00:00:00:00", "ifconnected": true, "iftype": "interface type", "ifbackingdevice": "interface backing device", "ifdvswitch_uuid": "interface switch uuid", "ifdvswitch_portgroup": "interface switch port group", "ifdvswitch_port": "interface switch port", "ifip": ["interface ip addresses"] }] Описание возвращаемых данных см. в ключе элемента данных ”vmware.vm.net.if.discovery”.
vmware.vm.net.if.discovery		
Выполняет обнаружение сетевых интерфейсов виртуальной машины.	{#IFNAME}	Имя сетевого интерфейса.
	{#IFDESC}	Описание интерфейса.
	{#IFMAC}	MAC-адрес интерфейса.
	{#IFCONNECTED}	Состояние подключения интерфейса (`false` - отключен; `true` - подключен).
	{#IFTYPE}	Тип интерфейса.
	{#IFBACKINGDEVICE}	Имя backing device.
	{#IFDVSWITCH.UUID}	Уникальный идентификатор vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Распределенная группа портов.
	{#IFDVSWITCH.PORT}	Порт vSphere Distributed Switch.
	”ifip”	Массив адресов интерфейса.
vmware.vm.vfs.dev.discovery		
Выполняет обнаружение дисковых устройств виртуальной машины.	{#DISKNAME}	Имя дискового устройства.
vmware.vm.vfs.fs.discovery		
Выполняет обнаружение файловых систем виртуальной машины.	{#FSNAME}	Имя файловой системы.

Примеры JSON для элементов данных VMware

Обзор В этом разделе приводится дополнительная информация о JSON-объектах, возвращаемых различными элементами данных VMware.

vmware.*.alarms.get Элементы данных **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** возвращают объекты JSON со следующей структурой (значения приведены в качестве примера):

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get Элементы данных **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** возвращают объекты JSON следующей структуры (значения предоставлены для примера):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "tag for cat OS type",
        "category": "OS type"
    },
    {
            "name": "SQL Server",
        "description": "tag for cat application name",
        "category": "application name"
        },
    {
```

```
            //
        }
    ]
}
```

vmware.hv.diskinfo.get Элемент данных **vmware.hv.diskinfo.get[]** возвращает объекты JSON следующей структуры (значения предоставлены для примера):

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
    "datastore_uuid": "",
    "operational_state": [
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    //
  }
]
```

vmware.dvswitch.fetchports.get Элемент данных **vmware.dvswitch.fetchports.get[]** возвращает объекты JSON следующей структуры (значения предоставлены для примера):

```
{
    "FetchDVPortsResponse":
    {
        "returnval": [
            {
                "key": "0",
                "dvsUuid": "50 36 6a 24 25 c0 10 9e-05 4a f6 ea 4e 3d 09 88",
                "portgroupKey": "dvportgroup-2023",
                "proxyHost":
                {
                    "@type": "HostSystem",
                    "#text": "host-2021"
                },
                "connectee":
                {
                    "connectedEntity":
                    {
                        "@type": "HostSystem",
                        "#text": "host-2021"
                    },
                    "nicKey": "vmnic0",
                    "type": "pnic"
                },
                "conflict": "false",
                "state":
                {
                    "runtimeInfo":
                    {
                        "linkUp": "true",
                        "blocked": "false",
                        "vlanIds":
```

```
                        {
                            "start": "0",
                            "end": "4094"
                        },
                        "trunkingMode": "true",
                        "linkPeer": "vmnic0",
                        "macAddress": "00:00:00:00:00:00",
                        "statusDetail": null,
                        "vmDirectPathGen2Active": "false",
                        "vmDirectPathGen2InactiveReasonOther": "portNptIncompatibleConnectee"
                    },
                    "stats":
                    {
                        "packetsInMulticast": "2385470",
                        "packetsOutMulticast": "45",
                        "bytesInMulticast": "309250248",
                        "bytesOutMulticast": "5890",
                        "packetsInUnicast": "155601537",
                        "packetsOutUnicast": "113008658",
                        "bytesInUnicast": "121609489384",
                        "bytesOutUnicast": "47240279759",
                        "packetsInBroadcast": "1040420",
                        "packetsOutBroadcast": "7051",
                        "bytesInBroadcast": "77339771",
                        "bytesOutBroadcast": "430392",
                        "packetsInDropped": "0",
                        "packetsOutDropped": "0",
                        "packetsInException": "0",
                        "packetsOutException": "0"
                    }
                },
                "connectionCookie": "1702765133",
                "lastStatusChange": "2022-03-25T14:01:11Z",
                "hostLocalPort": "false"
            },
            {
                //
            }
        ]
    }
}
```


vmware.hv.hw.sensors.get Элемент данных **vmware.hv.hw.sensors.get[]** возвращает объекты JSON следующей структуры (значения предоставлены для примера):


```
{
    "val":
        {
            "@type": "HostHardwareStatusInfo",
            "storageStatusInfo": [
                {
                    "name": "Intel Corporation HD Graphics 630 #2",
                "status":
                        {
                            "label": "Unknown",
                    "summary": "Cannot report on the current status of the physical element",
                    "key": "Unknown"
                }
            },
            {
                    "name": "Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller #20"
                "status":
                        {
```

```
                    "label": "Unknown",
                    "summary": "Cannot report on the current status of the physical element",
                    "key": "Unknown"
                }
            },
            {
                //
            }
        ]
        }
}
```

vmware.hv.sensors.get Элемент данных **vmware.hv.sensors.get[]** возвращает объекты JSON следующей структуры (значения предоставлены для примера):

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                //
            }
        ]
        }
}
```

vmware.vm.snapshot.get Если существуют снапшоты, то элемент данных **vmware.snapshot.get[]** возвращает объекты JSON следующей структуры (значения предоставлены для примера):

```
{
  "snapshot": [
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
```

```
      "size": 5755795171,
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```


При отсутствии снапшотов элемент данных **vmware.snapshot.get[]** возвращает объект JSON с пустыми значениями:


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


Пример настройки мониторинга VMware

Обзор

В следующем примере описано, как настроить Zabbix для мониторинга виртуальных машин VMware. Это включает в себя:

- создание узла сети, представляющего среду виртуализации VMware;
- создание правила низкоуровневого обнаружения, которое обнаруживает виртуальные машины в вашей среде виртуализации VMware;
- создание прототипа узла сети, на основе которого Zabbix будет генерировать реальные узлы сети для виртуальных машин, обнаруженных правилом низкоуровневого обнаружения.

Предварительные требования

> **Note:**
> В этом примере не рассматривается настройка VMware. Предполагается, что VMware уже настроена.

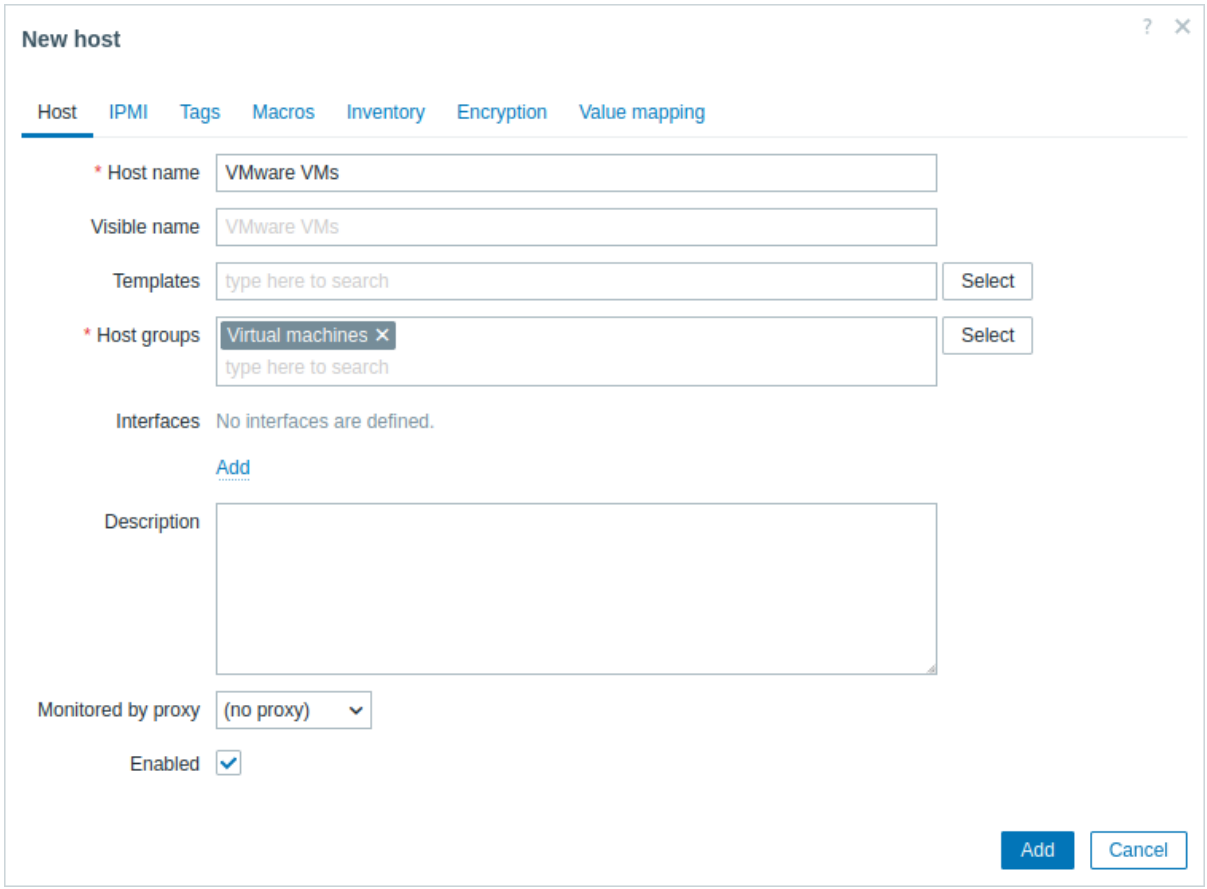
Перед продолжением установите параметр `StartVMwareCollectors` в файле конфигурации сервера Zabbix в значение 2 или больше (значение по умолчанию — 0).

Создание узла сети

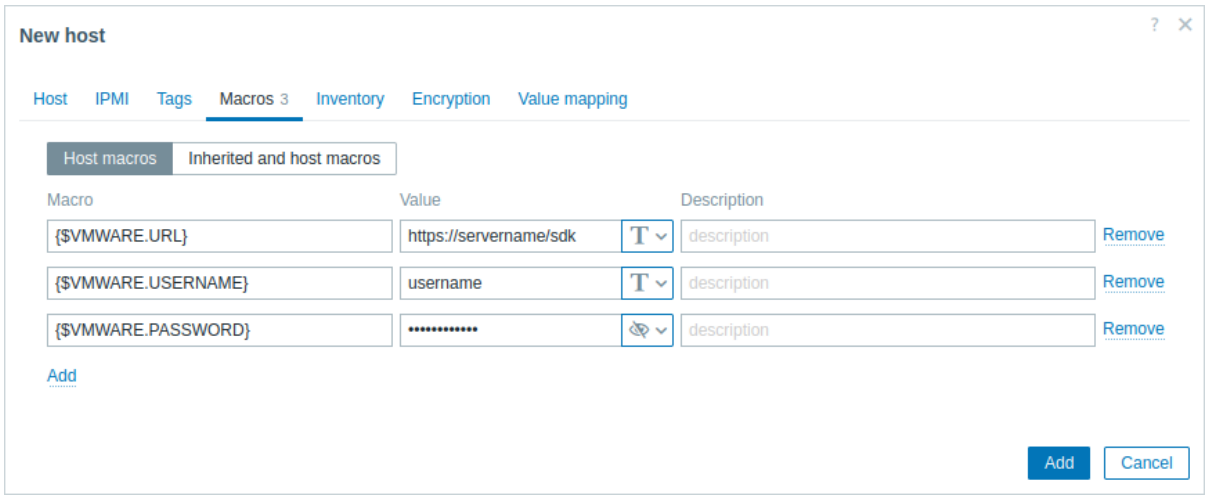
1. Перейдите в *Сбор данных* > *Узлы сети*.

2. Создайте узел сети:

- В поле *Host name* введите имя узла сети (например, ”VMware VMs”).
- В поле *Host groups* введите или выберите группу узлов сети (например, ”Virtual machines”).



- На вкладке *Macros* задайте следующие макросы узла сети:
  - {$VMWARE.URL} - URL SDK службы VMware (гипервизор ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - имя пользователя службы VMware
  - {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} службы VMware



3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять вашу среду VMware.

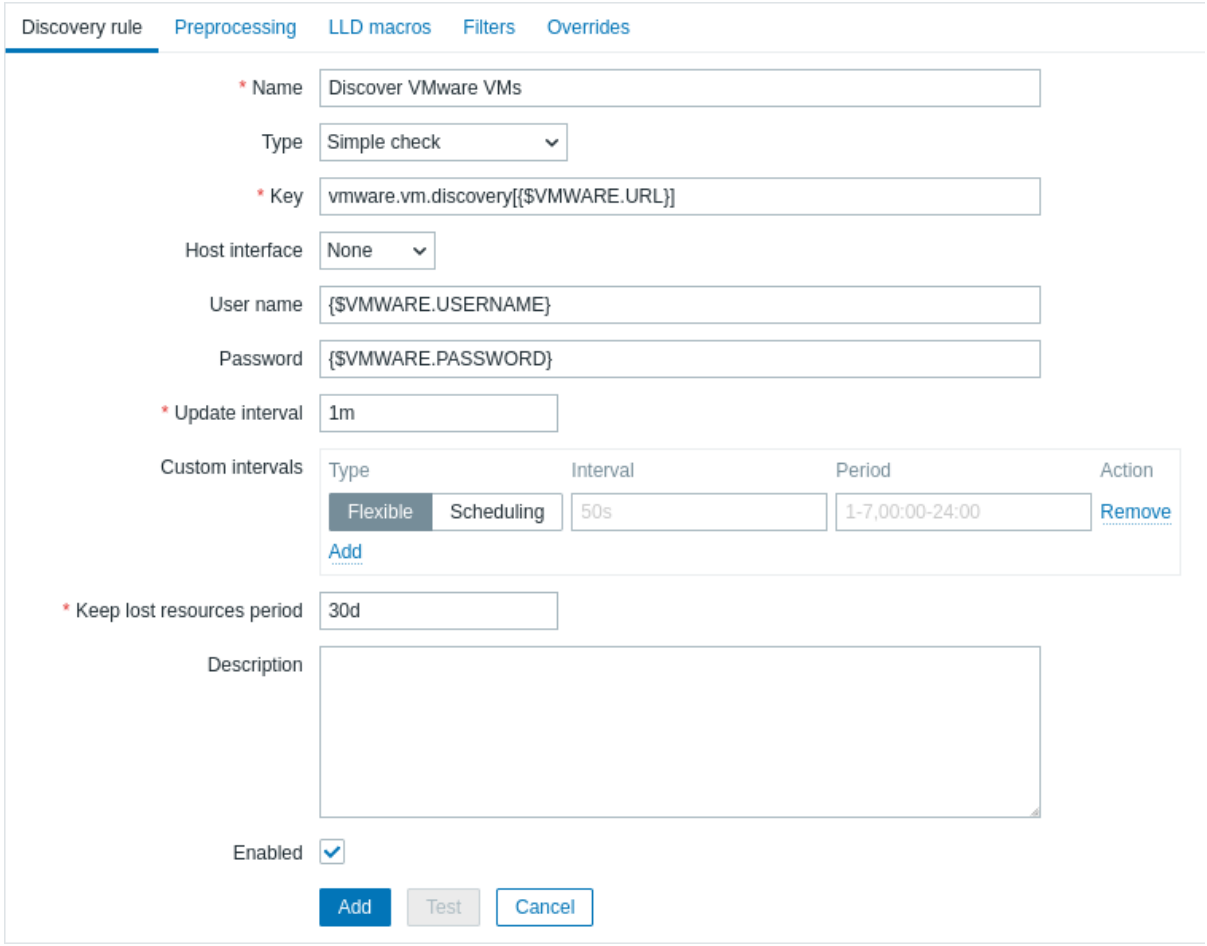
Создание правила низкоуровневого обнаружения

1. Нажмите *Обнаружение (Discovery)* напротив созданного узла сети, чтобы перейти к списку правил низкоуровневого обнаружения этого узла сети.

2. Создайте правило низкоуровневого обнаружения:

- В поле *Имя (Name)* введите имя правила низкоуровневого обнаружения (например, «Discover VMware VMs»).
- В поле *Тип (Type)* выберите «Простая проверка (Simple check)».

- В поле *Ключ (Key)* введите встроенный ключ элемента данных для обнаружения виртуальных машин VMware: `vmware.vm.discovery[{$VMWARE.URL}]`
- В полях *Имя пользователя (User name)* и *Пароль (Password)* введите соответствующие макросы, ранее настроенные на узле сети.



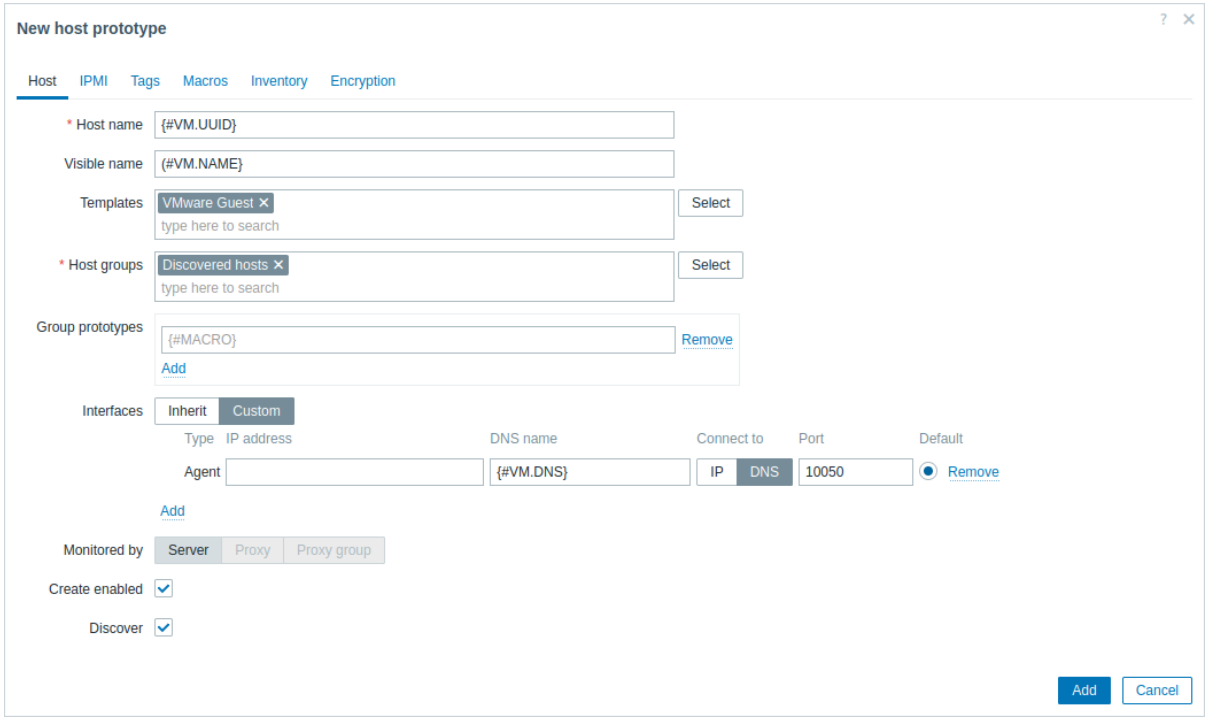
3. Нажмите кнопку *Добавить (Add)* для добавления правила низкоуровневого обнаружения. Это правило низкоуровневого обнаружения будет обнаруживать виртуальные машины в вашей среде виртуализации VMware.

Создание прототипа узла сети

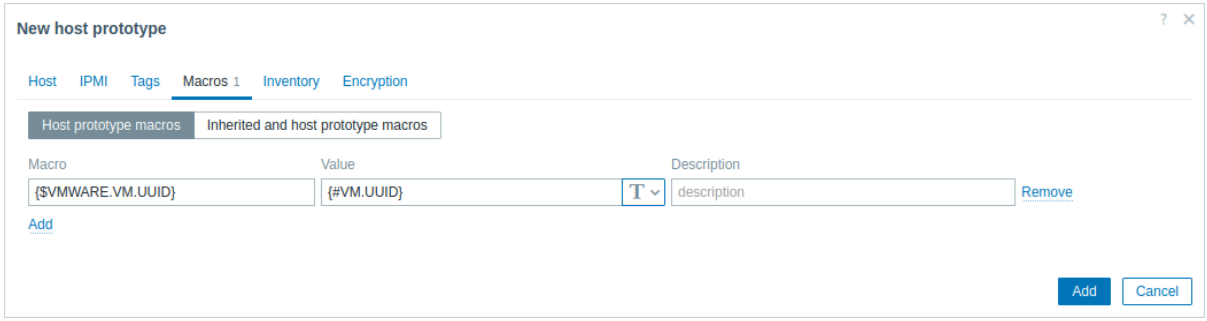
1. В списке правил низкоуровневого обнаружения напротив ранее созданного правила низкоуровневого обнаружения нажмите *Прототипы узлов сети (Host prototypes)*.

2. Создайте прототип узла сети. Поскольку прототипы узла сети — это заготовки для создания узлов сети посредством правил низкоуровневого обнаружения, большинство полей будут содержать макросы низкоуровневого обнаружения. Это гарантирует, что узлы сети будут созданы со свойствами на основе содержимого, полученного ранее созданным правилом низкоуровневого обнаружения.

- В поле *Имя узла сети (Host name)* введите макрос {#VM.UUID}.
- В поле *Видимое имя (Visible name)* введите макрос {#VM.NAME}.
- В поле *Шаблоны (Templates)* введите или выберите шаблон «VMware Guest». Это шаблон содержит элементы данных VMware и правила обнаружения для мониторинга состояния питания виртуальной машины, использования CPU, использования памяти, сетевые устройства и т.д.
- В поле *Группы узлов сети (Host groups)* введите или выберите группу узлов сети (например, «Discovered hosts»).
- В поле *Интерфейсы (Interfaces)* добавьте пользовательский интерфейс узла сети. Затем введите макрос {#VM.DNS} в поле *DNS имя (DNS name)* или введите макрос {#VM.IP} в поле *IP адрес (IP address)*. В качестве альтернативы, если виртуальные машины вашей среды VMware имеют несколько интерфейсов, перейдите к разделу *Расширенная настройка интерфейса узла сети*. Настройка пользовательского интерфейса узла сети нужна для правильного функционирования шаблона *VMware Guest*.



- На вкладке *Макросы (Macros)* укажите макрос {$VMWARE.VM.UUID} со значением {#VM.UUID}. Это необходимо для правильного функционирования шаблона *VMware Guest*, который использует данный макрос как пользовательский макрос уровня узла сети в параметрах элементов данных (например, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

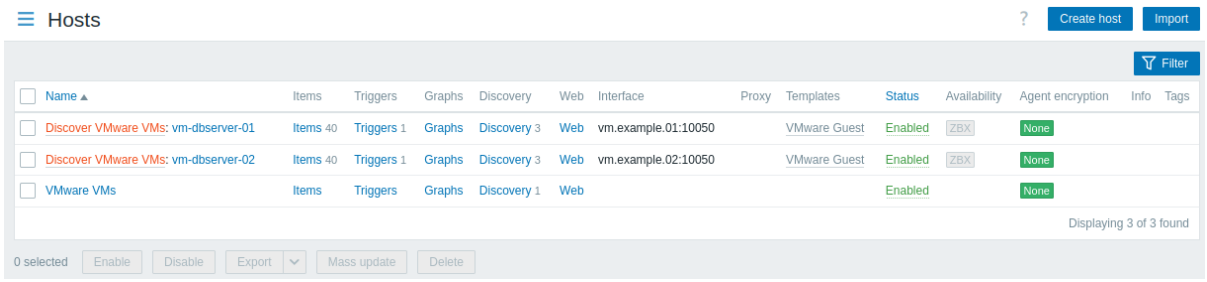


3. Нажмите кнопку *Добавить (Add)*, чтообы создать прототип узла сети. Этот прототип узла сети будет использоваться для создания узлов сети для виртуальных машин, обнаруженных ранее созданным правилом низкоуровневого обнаружения.

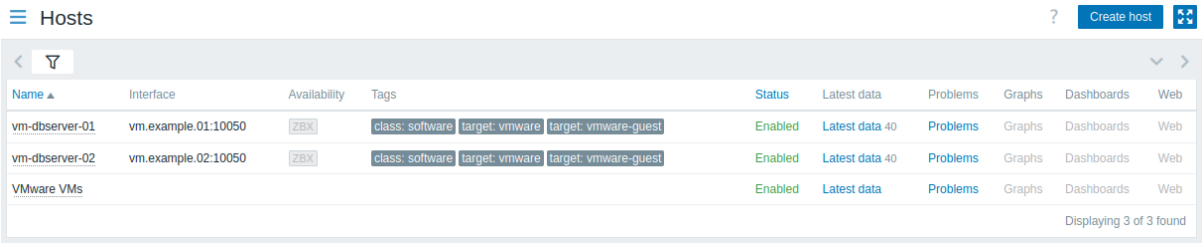
Просмотр узлов сети и метрик

После создания прототипа узла сети правило низкоуровневого обнаружения создаст узлы сети для обнаруженных виртуальных машин VMware, и Zabbix начнет их мониторинг. Обратите внимание, что обнаружение и создание узлов сети также можно выполнить вручную, если это необходимо.

Чтобы просмотреть созданные узлы сети, перейдите в раздел меню *Сбор данных > Узлы сети*.



Чтобы просмотреть собранные метрики, перейдите в раздел меню *Мониторинг > Узлы сети* и нажмите *Последние данные* для одного из узлов сети.



Расширенная настройка интерфейса узла сети

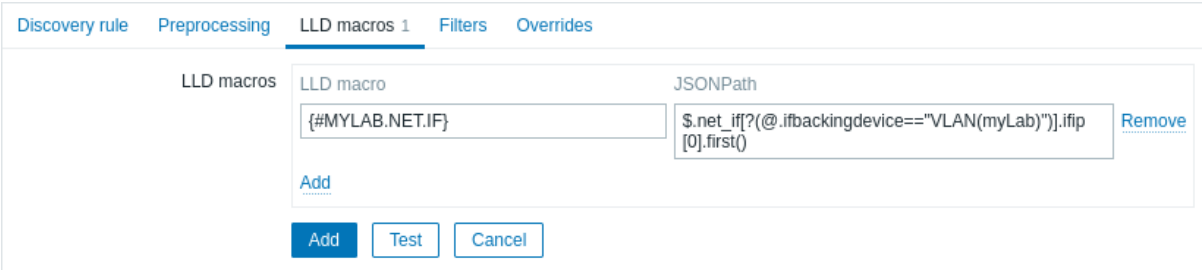
Ключ элемента данных `vmware.vm.discovery[{$VMWARE.URL}]`, настроенный в разделе *Создание правила низкоуровневого обнаружения*, возвращает данные интерфейсов узлов сети в поле "net_if":

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

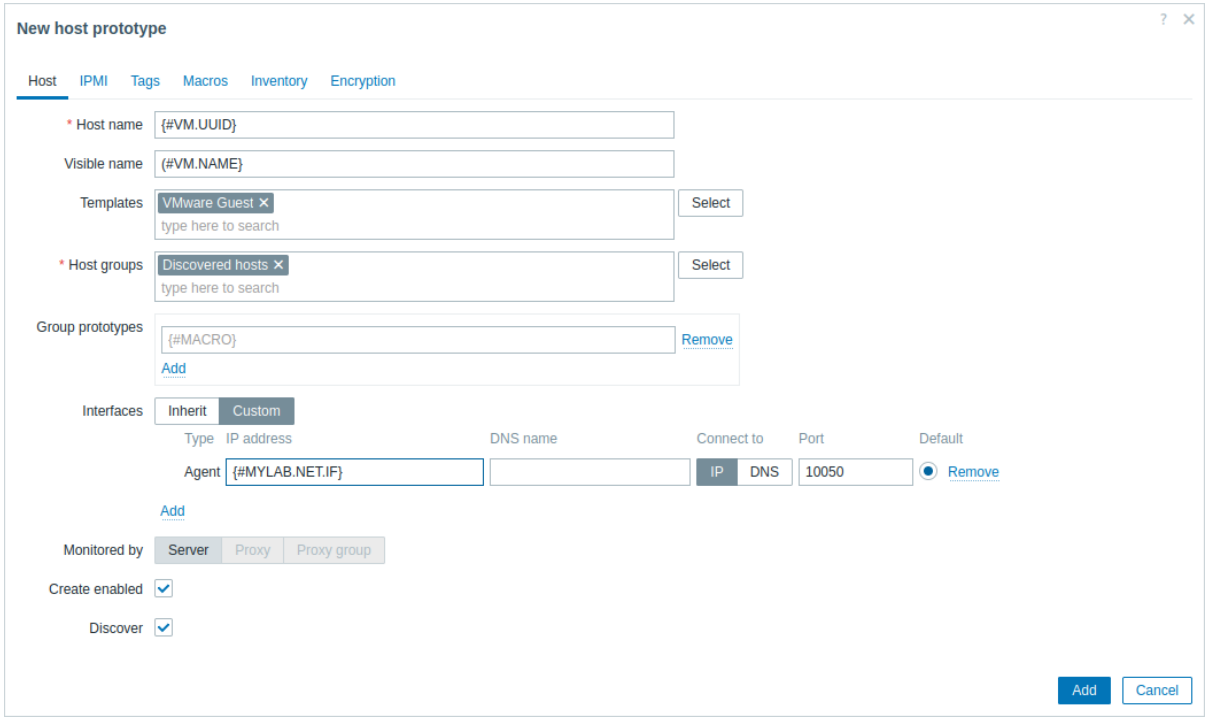
Эти данные могут быть использованы, чтобы настроить пользовательский интерфейс узла сети.

1. При создании правила низкоуровневого обнаружения дополнительно настройте макрос низкоуровневого обнаружения. На вкладке *LLD макросы (LLD macros)* создайте пользовательский LLD макрос со значением JSONPath. Например:

- `{#MYLAB.NET.IF}` — `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. При создании прототипа узла сети добавьте пользовательский интерфейс узла сети и введите LLD макрос в поле *DNS имя (DNS name)* или *IP адрес (IP address)*.



Веб-мониторинг

Обзор С помощью Zabbix вы можете проверять несколько аспектов доступности веб-сайтов.

Attention:
Для выполнения веб-мониторинга сервер Zabbix должен быть изначально настроен с поддержкой cURL (libcurl).

Чтобы включить веб-мониторинг, необходимо определить веб-сценарии. Веб-сценарий состоит из одного или нескольких HTTP-запросов или "шагов". Шаги периодически выполняются сервером Zabbix в заранее определенном порядке. Если узел сети контролируется через прокси, шаги выполняются прокси.

Веб-сценарии привязываются к узлам сети/шаблонам так же, как элементы данных, триггеры и т. д. Это означает, что веб-сценарии также можно создавать на уровне шаблона, а затем применять к нескольким узлам сети за один раз.

В любом веб-сценарии собирается следующая информация:

- средняя скорость загрузки в секунду для всех шагов всего сценария
- номер шага, на котором произошел сбой
- последнее сообщение об ошибке

В любом шаге веб-сценария собирается следующая информация:

- скорость загрузки в секунду
- время ответа
- код ответа

Подробнее см. элементы данных веб-мониторинга.

Данные, собранные при выполнении веб-сценариев, сохраняются в базе данных. Эти данные автоматически используются для графиков, триггеров и уведомлений.

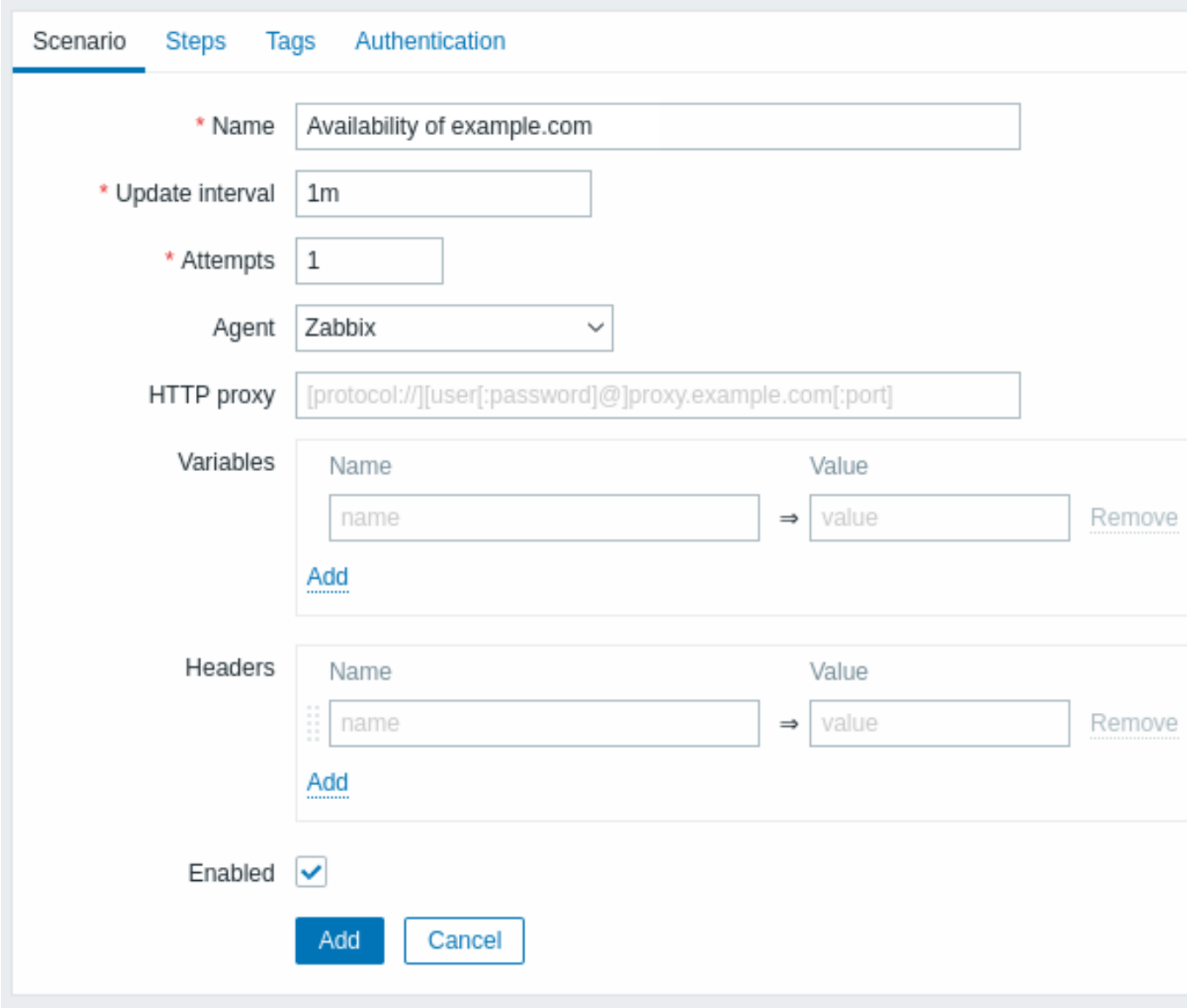
Zabbix также может проверять, содержит ли полученная HTML-страница заранее определенную строку. Он может выполнять имитацию входа в систему и следовать по пути имитируемых щелчков мышью на странице.

Веб-мониторинг Zabbix поддерживает HTTP и HTTPS. При запуске веб-сценария Zabbix при необходимости будет следовать перенаправлениям (см. параметр *Follow redirects* ниже). Максимальное число перенаправлений жестко задано и равно 10 (с использованием опции cURL CURLOPT_MAXREDIRS). Все cookies сохраняются на протяжении выполнения одного сценария.

Настройка веб-сценария Чтобы настроить веб-сценарий:

1. Перейдите в раздел: *Сбор данных* > *Узлы сети* (или *Шаблоны*).
2. Нажмите *Веб* в строке узла сети/шаблона.
3. Справа нажмите *Создать веб-сценарий* (или имя сценария, чтобы изменить существующий сценарий).
4. Введите параметры сценария в форме.

Вкладка **Сценарий** позволяет настроить общие параметры веб-сценария.



Все обязательные поля ввода отмечены красной звездочкой.

Параметры сценария:

Параметр	Описание
Имя	Уникальное имя сценария. Поддерживаются пользовательские макросы. *Обратите внимание*, что при использовании пользовательских макросов эти макросы останутся нераскрытыми в именах элементов данных веб-мониторинга.
Интервал обновления	Как часто будет выполняться сценарий. Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. *Обратите внимание*, что если используется пользовательский макрос и его значение изменяется (например, 5m → 30s), следующая проверка будет выполнена в соответствии с предыдущим значением (в приведенном примере — позже). Новые веб-сценарии будут проверены в течение 60 секунд после создания.

Параметр	Описание
Попытки	Количество попыток выполнения шагов веб-сценария. В случае сетевых проблем (тайм-аут, отсутствие соединения и т. д.) Zabbix может повторить выполнение шага несколько раз. Указанное значение применяется одинаково к каждому шагу сценария. Можно указать до 10 попыток, значение по умолчанию — 1. *Примечание*: Zabbix не будет повторять шаг из-за неправильного кода ответа или несоответствия требуемой строки.
Агент	Выберите клиентский агент. Zabbix будет имитировать выбранный браузер. Это полезно, если веб-сайт возвращает разное содержимое для разных браузеров. В этом поле можно использовать пользовательские макросы.
HTTP-прокси	Можно указать используемый HTTP-прокси в формате `[protocol://][username[:password]@]proxy.example.com[:port]`. Это устанавливает параметр cURL CURLOPT_PROXY. Необязательный префикс `protocol://` можно использовать для указания альтернативных протоколов прокси (поддержка префикса протокола добавлена в cURL 7.21.7). Если протокол не указан, прокси будет считаться HTTP-прокси. По умолчанию используется порт 1080. Если прокси указан, он переопределяет переменные окружения, связанные с прокси, например http_proxy и HTTPS_PROXY. Если прокси не указан, он не переопределяет переменные окружения, связанные с прокси. Введенное значение передается «как есть», без проверки корректности. Также можно указать адрес SOCKS-прокси. Если указать неправильный протокол, соединение завершится ошибкой, а элемент данных станет неподдерживаемым. *Обратите внимание*, что для HTTP-прокси поддерживается только простая аутентификация. В этом поле можно использовать пользовательские макросы.
Переменные	Переменные, которые можно использовать в шагах сценария (URL, переменные post-запроса). Они имеют следующий формат: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (см. функции макросов) Например: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Затем на эти макросы можно ссылаться в шагах как {username}, {password}, {hostid} и т. д. Zabbix автоматически заменит их фактическими значениями. Обратите внимание, что для переменных с `regex:` необходим шаг, получающий значение регулярного выражения, поэтому извлеченное значение можно применить только к следующему шагу. Если часть значения начинается с `regex:`, то следующая за ним часть рассматривается как регулярное выражение, выполняющее поиск на веб-странице; если совпадение найдено, оно сохраняется в переменной. Для извлечения совпавшего значения должна присутствовать как минимум одна подгруппа. Поддерживаются пользовательские макросы и `{HOST.*}` макросы. Переменные автоматически кодируются в URL при использовании в полях запроса или данных формы для post-переменных, но при использовании в необработанном post-запросе или непосредственно в URL их необходимо кодировать в URL вручную.

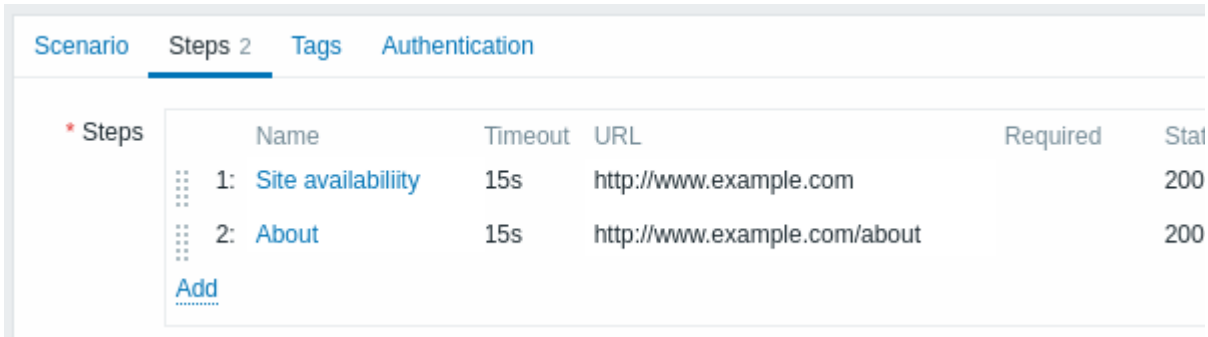
Параметр	Описание
Заголовки	HTTP-заголовки используются при выполнении запроса. Можно использовать заголовки по умолчанию и пользовательские заголовки. Заголовки назначаются с использованием настроек по умолчанию в зависимости от типа агента, выбранного в раскрывающемся списке на уровне сценария, и применяются ко всем шагам, если только они не определены отдельно на уровне шага. **Обратите внимание, что определение заголовка на уровне шага автоматически отменяет все ранее определенные заголовки, за исключением заголовка по умолчанию, назначенного выбором «User-Agent» в раскрывающемся списке на уровне сценария.** Однако даже заголовок «User-Agent» по умолчанию можно переопределить, указав его на уровне шага. Чтобы отменить заголовок на уровне сценария, на уровне шага необходимо указать имя заголовка без значения. Заголовки следует указывать с использованием того же синтаксиса, в котором они представлены в протоколе HTTP, при необходимости применяя дополнительные возможности, поддерживаемые параметром cURL CURLOPT_HTTPHEADER. Например: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Поддерживаются пользовательские макросы и `{HOST.*}` макросы.
Включен	Сценарий активен, если этот флажок установлен; в противном случае он отключен.

Обратите внимание, что при редактировании существующего сценария в форме доступны две дополнительные кнопки:

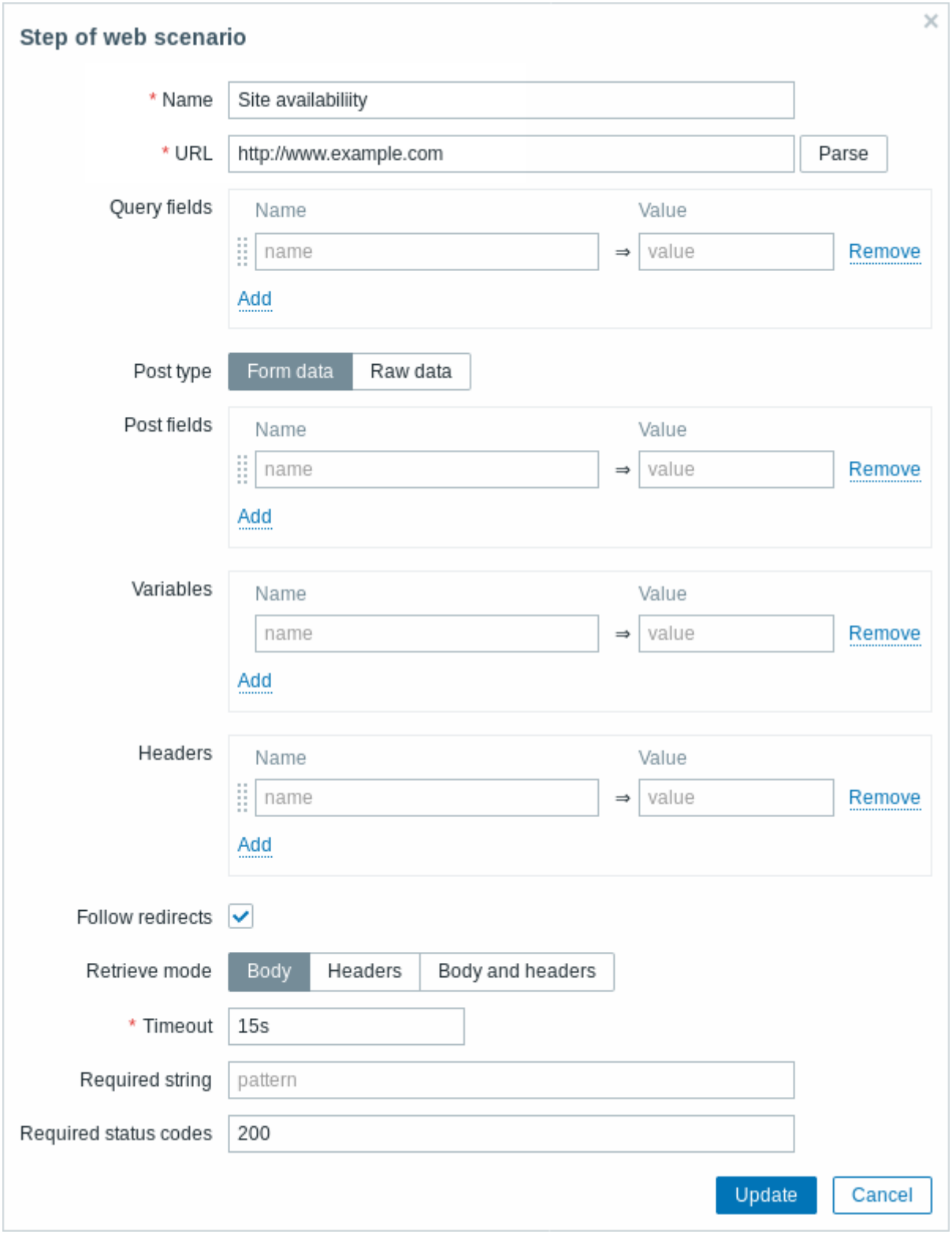
Clone	Создать другой сценарий на основе свойств существующего сценария.
Clear history and trends	Удалить данные истории и трендов для сценария. После удаления данных сервер немедленно выполнит сценарий.

Note:
Если поле *HTTP-прокси* оставить пустым, для использования HTTP-прокси можно задать переменные окружения, связанные с прокси.
Для HTTP-проверок задайте переменную окружения `http_proxy` для пользователя сервера Zabbix. Например, `http_proxy=http://proxy_ip:proxy_port`.
Для HTTPS-проверок задайте переменную окружения `HTTPS_PROXY`. Например, `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Дополнительные сведения можно получить, выполнив команду оболочки: `# man curl`.

Вкладка **Шаги** позволяет настроить шаги веб-сценария. Чтобы добавить шаг веб-сценария, нажмите *Добавить* в блоке *Шаги*.



Note:
Секретные пользовательские макросы нельзя использовать в URL, поскольку они будут преобразованы в "******".



Настройка шагов

Параметры шага:

Параметр	Описание
Имя	Уникальное имя шага. Поддерживаются пользовательские макросы. *Обратите внимание*, что если используются пользовательские макросы, эти макросы останутся неразрешенными в именах элементов данных веб-мониторинга.

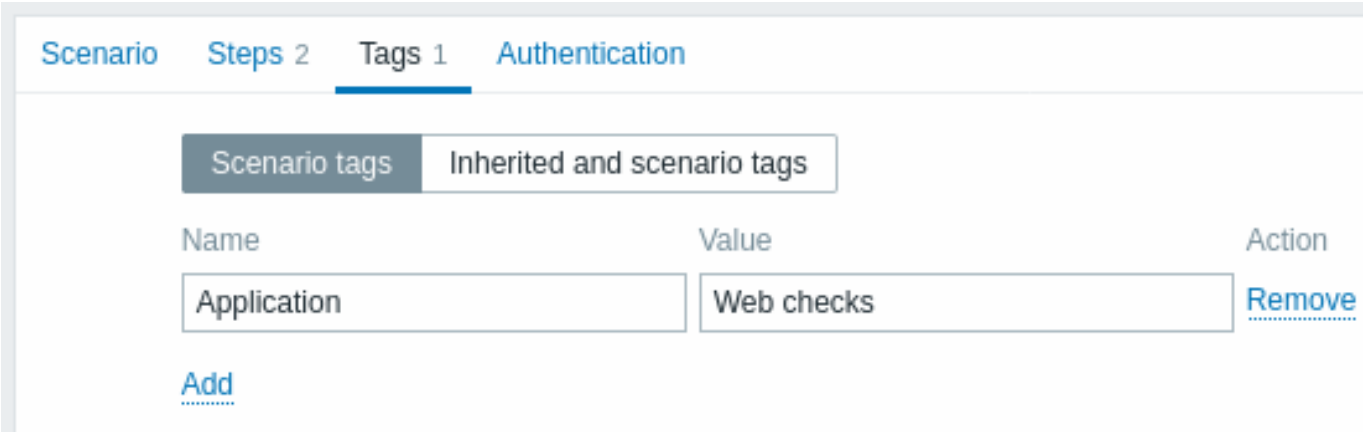
Параметр	Описание
URL	URL-адрес для подключения и получения данных. Например: https://www.example.com http://www.example.com/download Доменные имена можно указывать с использованием символов Unicode. При выполнении шага веб-сценария они автоматически преобразуются из punycode в ASCII. Кнопка *Разобрать* позволяет отделить необязательные поля запроса (например, `?key=value&limit=10`) от URL-адреса, переместив атрибуты и значения в *Поля запроса* для автоматического URL-кодирования. В URL можно использовать переменные с синтаксисом {macro}. Переменные можно кодировать в URL вручную с помощью синтаксиса {{macro}.urlencode()}. Поддерживаются пользовательские макросы и макросы `{HOST.*}`. В целях безопасности не включайте реальные имена пользователей или пароли в URL-адреса. Вместо этого используйте пользовательские макросы, чтобы не раскрывать учетные данные. Ограничено 2048 символами.
Поля запроса	Переменные HTTP GET для URL-адреса. Указываются в виде пар атрибут-значение. Значения автоматически кодируются в URL. Значения переменных сценария, пользовательских макросов или макросов `{HOST.*}` сначала разрешаются, а затем автоматически кодируются в URL. Использование синтаксиса {{macro}.urlencode()} приведет к двойному URL-кодированию. Поддерживаются пользовательские макросы и макросы `{HOST.*}`.
Post	Переменные HTTP POST. В режиме **Данные формы** указываются в виде пар атрибут-значение. Значения автоматически кодируются в URL. Значения переменных сценария, пользовательских макросов или макросов `{HOST.*}` сначала разрешаются, а затем автоматически кодируются в URL. В режиме **Необработанные данные** атрибуты и значения отображаются в одной строке и объединяются символом **&**. Необработанные значения можно кодировать или декодировать в URL вручную с помощью синтаксиса {{macro}.urlencode()} или {{macro}.urldecode()}. Например: id=2345&userid={user} Если `{user}` определена как переменная веб-сценария, при выполнении шага она будет заменена своим значением. Если переменную необходимо закодировать в URL, замените `{user}` на {{user}.urlencode()}. Поддерживаются пользовательские макросы и макросы `{HOST.*}`.
Переменные	Переменные уровня шага, которые можно использовать для функций GET и POST. Указываются в виде пар атрибут-значение. Переменные уровня шага имеют приоритет над переменными уровня сценария или переменными предыдущего шага. Однако значение переменной уровня шага влияет только на следующий шаг, а не на текущий. Они имеют следующий формат: **{macro}**=value **{macro}**=regex:<regular expression> Дополнительную информацию см. в описании переменных на уровне сценария. Переменные автоматически кодируются в URL при использовании в полях запроса или данных формы для переменных post, но при использовании в необработанных данных post или непосредственно в URL их необходимо кодировать в URL вручную.
Заголовки	Пользовательские HTTP-заголовки, отправляемые при выполнении запроса. Указываются в виде пар атрибут-значение. Заголовок, определенный на уровне шага, будет использоваться для соответствующего шага. **Обратите внимание, что определение заголовка на уровне шага автоматически отменяет все ранее определенные заголовки, за исключением заголовка по умолчанию, назначенного путем выбора 'User-Agent' из раскрывающегося списка на уровне сценария.** Однако даже заголовок 'User-Agent' по умолчанию можно переопределить, указав его на уровне шага. Например, если назначить имя заголовку, но не указать значение, заголовок по умолчанию на уровне сценария будет отменен. Поддерживаются пользовательские макросы и макросы `{HOST.*}`. Это устанавливает опцию cURL `CURLOPT_HTTPHEADER`.

Параметр	Описание
Следовать перенаправлениям	Установите флажок, чтобы следовать HTTP-перенаправлениям. Это устанавливает опцию cURL CURLOPT_FOLLOWLOCATION.
Режим получения	Выберите режим получения: **Тело** — получать только тело HTTP-ответа **Заголовки** — получать только заголовки HTTP-ответа **Тело и заголовки** — получать тело и заголовки HTTP-ответа
Тайм-аут	Zabbix не будет тратить на обработку URL больше заданного времени (от одной секунды до одного часа). Фактически этот параметр определяет максимальное время установления соединения с URL и максимальное время выполнения HTTP-запроса. Поэтому Zabbix не будет тратить на шаг более **2 x Тайм-аут** секунд. Поддерживаются суффиксы времени, например 30s, 1m, 1h. Поддерживаются пользовательские макросы.
Обязательная строка	Требуемый шаблон регулярного выражения. Если полученное содержимое (HTML) не соответствует требуемому шаблону, шаг завершится ошибкой. Если поле пустое, проверка обязательной строки не выполняется. Например: Homepage of Zabbix Welcome.*admin *Примечание*: в этом поле не поддерживаются ссылки на регулярные выражения, созданные в веб-интерфейсе Zabbix. Поддерживаются пользовательские макросы и макросы `{HOST.*}`.
Обязательные коды состояния	Список ожидаемых кодов состояния HTTP. Если Zabbix получает код, отсутствующий в списке, шаг завершится ошибкой. Если поле пустое, проверка кодов состояния не выполняется. Например: 200,201,210-299 Поддерживаются пользовательские макросы.

Note:
Любые изменения в шагах веб-сценария будут сохранены только после сохранения всего сценария.

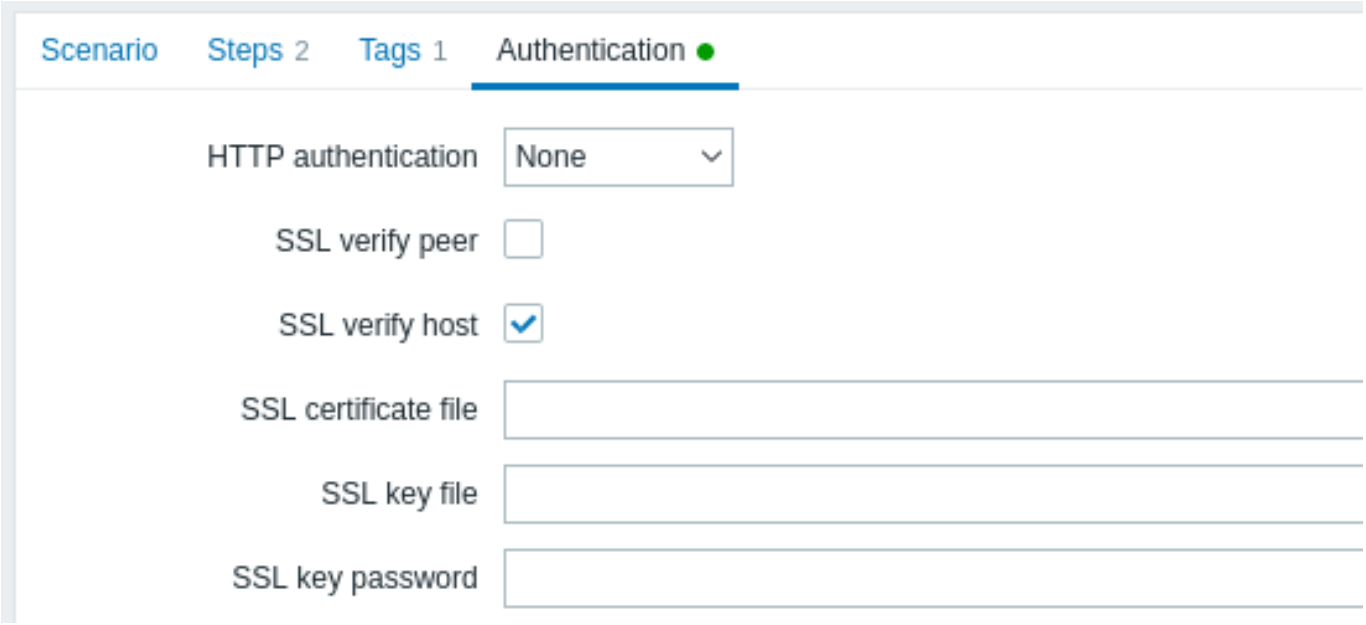
См. также пример из реальной практики настройки шагов веб-мониторинга.

Настройка тегов Вкладка **Теги** позволяет определить теги на уровне сценария.



Теги позволяют фильтровать веб-сценарии и элементы данных веб-мониторинга.

Настройка аутентификации Вкладка **Аутентификация** позволяет настроить параметры аутентификации сценария. Зеленая точка рядом с названием вкладки указывает, что включен какой-либо тип HTTP-аутентификации.



Параметры аутентификации:

Параметр	Описание
HTTP-аутентификация	Выберите вариант аутентификации: **Нет** — аутентификация не используется; **Basic** — используется базовая аутентификация; **NTLM** — используется аутентификация NTLM (Windows NT LAN Manager); **Kerberos** — используется аутентификация Kerberos (см. также: Настройка Kerberos с Zabbix); **Digest** — используется дайджест-аутентификация.
Пользователь	Введите имя пользователя (до 255 символов). Это поле доступно, если для *HTTP-аутентификации* выбрано значение Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы.
Пароль	Введите пароль пользователя (до 255 символов). Это поле доступно, если для *HTTP-аутентификации* выбрано значение Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы.
Проверять SSL-сертификат узла сети	Установите флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера проверяется по сертификатам доверенных центров сертификации (CA) из общесистемного расположения CA. Расположение файлов CA можно переопределить с помощью параметра конфигурации сервера или прокси Zabbix SSLCALocation. Устанавливает параметр cURL CURLOPT_SSL_VERIFYPEER.
Проверять имя узла сети SSL	Установите флажок, чтобы проверить соответствие поля *Common Name* или *Subject Alternate Name* в сертификате веб-сервера. Устанавливает параметр cURL CURLOPT_SSL_VERIFYHOST.
Файл SSL-сертификата	Имя файла SSL-сертификата, используемого для аутентификации клиента. Файл сертификата должен быть в формате PEM[1]. Если файл сертификата также содержит закрытый ключ, оставьте поле *Файл SSL-ключа* пустым. Если ключ зашифрован, укажите пароль в поле *Пароль SSL-ключа*. Каталог, содержащий этот файл, задается параметром конфигурации сервера или прокси Zabbix SSLCertLocation. В этом поле можно использовать макросы `HOST.*` и пользовательские макросы. Устанавливает параметр cURL CURLOPT_SSLCERT.
Файл SSL-ключа	Имя файла закрытого SSL-ключа, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM[1]. Каталог, содержащий этот файл, задается параметром конфигурации сервера или прокси Zabbix SSLKeyLocation. В этом поле можно использовать макросы `HOST.*` и пользовательские макросы. Устанавливает параметр cURL CURLOPT_SSLKEY.
Пароль SSL-ключа	Пароль файла закрытого SSL-ключа. В этом поле можно использовать пользовательские макросы. Устанавливает параметр cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix поддерживает сертификаты и файлы закрытых ключей только в формате PEM. Если данные сертификата и закрытого ключа находятся в файле формата PKCS #12 (обычно с расширением *.p12 или *.pfx), можно создать из него файл PEM с помощью следующих команд:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

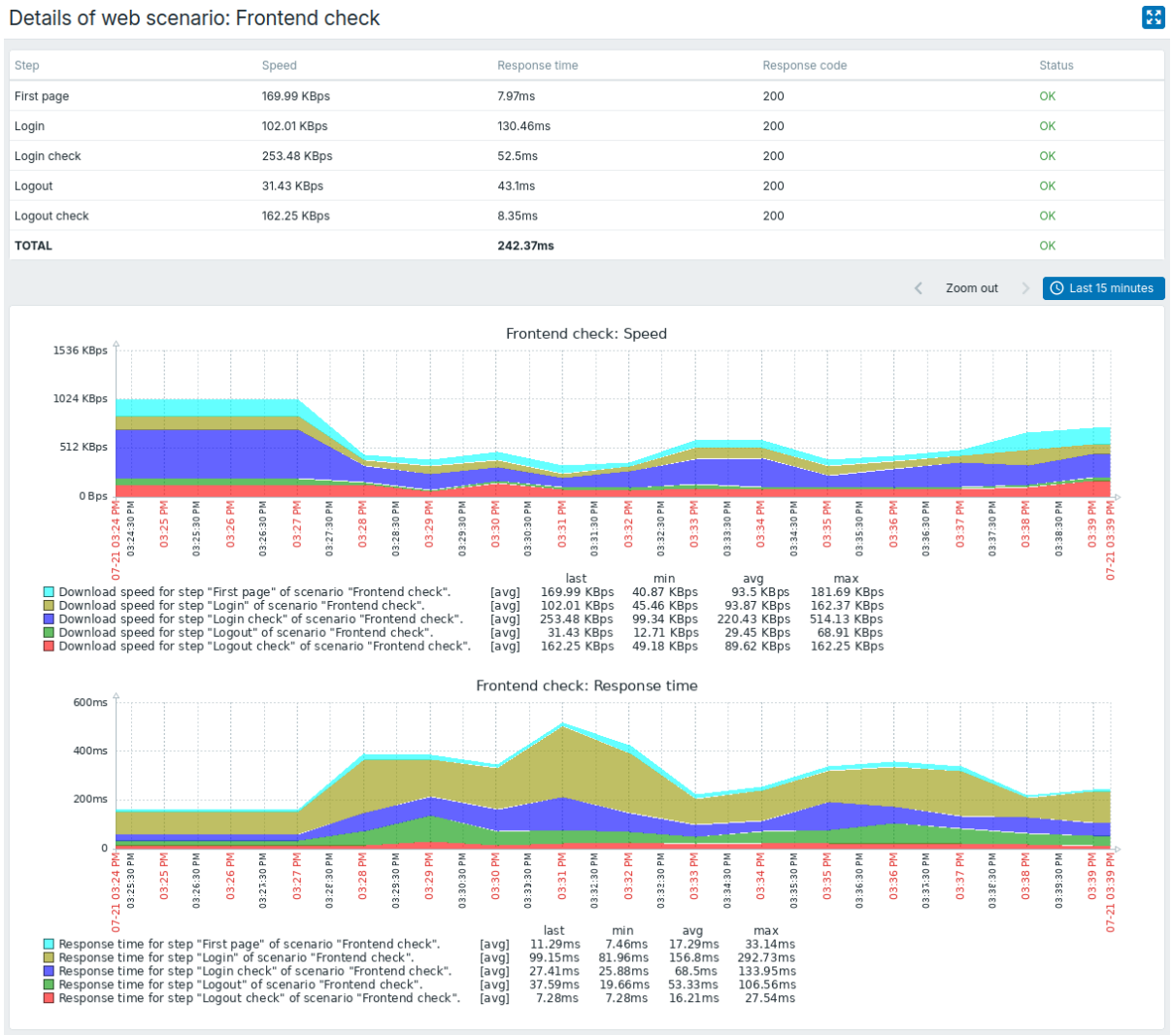
Note:
Сервер Zabbix применяет изменения сертификатов без перезапуска.

Note:
Если сертификат клиента и закрытый ключ находятся в одном файле, укажите его в поле «Файл SSL-сертификата», а поле «Файл SSL-ключа» оставьте пустым. Сертификат и ключ по-прежнему должны быть в формате PEM. Объединить сертификат и ключ можно следующим образом:

```
cat client.crt client.key > client.pem
```

Отображение Чтобы просмотреть веб-сценарии, настроенные для узла сети:

1. Перейдите в раздел *Мониторинг* > *Узлы сети*.
2. Найдите узел сети в списке и нажмите ссылку *Web* в последнем столбце.
3. Нажмите имя сценария, чтобы просмотреть подробную информацию.



Обзор веб-сценариев также можно отобразить в разделе *Панели мониторинга* с помощью виджета веб-мониторинга.

Последние результаты выполнения веб-сценария доступны в разделе *Мониторинг* > *Последние данные*.

Расширенный мониторинг Иногда необходимо записывать полученное содержимое HTML-страницы. Это особенно полезно, если какой-либо шаг веб-сценария завершается сбоем.

Для этого предназначен уровень отладки 5 (trace).
Этот уровень можно задать в файлах конфигурации сервер и прокси или с помощью параметра управления во время выполнения (`-R log_level_increase="http poller,N"`, где N — номер процесса).
Следующие примеры демонстрируют, как можно включить расширенный мониторинг, если уже установлен уровень отладки 4:

```
# Increase log level of all http pollers:
zabbix_server -R log_level_increase="http poller"

# Increase log level of second http poller:
zabbix_server -R log_level_increase="http poller,2"
```

Если расширенный веб-мониторинг не требуется, его можно остановить с помощью параметра `-R log_level_decrease`.

Элементы данных веб-мониторинга

Обзор

При создании веб-сценариев некоторые новые элементы данных для выполнения мониторинга добавляются автоматически.

Все элементы данных наследуют теги веб-сценария.

Элементы данных сценария

Как только сценарий будет создан, Zabbix автоматически добавит сопутствующие элементы данных для мониторинга.

Элемент данных	Описание
Download speed for scenario <Scenario> (*Скорость загрузки сценария <Сценарий>*)	Этот элемент данных собирает информацию о скорости загрузки (байт в секунду) всего сценария, т.е. средний показатель для всех шагов. Ключ элемента данных: web.test.in[Сценарий,,bps] Тип: *Числовой (с плавающей точкой)*
Failed step of scenario <Scenario> (*Неуспешный шаг сценария <Сценарий>*)	Этот элемент данных отображает номер неуспешного шага в сценарии. Если все шаги в сценарии выполнены успешно, то возвращается 0. Ключ элемента данных: web.test.fail[Сценарий] Тип: *Числовой (целое)*
Last error message of scenario <Scenario> (*Последнее сообщение об ошибке для сценария <Сценарий>*)	Этот элемент данных возвращает текст последнего сообщения об ошибке в сценарии. Новое значение сохраняется только, если в сценарии есть неуспешный шаг. Если все шаги в сценарии выполнены успешно, то значение не сохраняется. Ключ элемента данных: web.test.error[Сценарий] Тип: *Символ*

Вместо «Сценарий» будет использовано реальное имя сценария.

Note:
Если имя сценария содержит пользовательские макросы, эти макросы останутся нераскрытыми в именах элементов данных веб-мониторинга.

 Если имя сценария начинается с двойной кавычки или содержит запятую либо квадратные скобки, оно будет корректно заключено в кавычки в ключах элементов данных. В других случаях дополнительные кавычки не добавляются.

Note:
Элементы данных веб-мониторинга добавляются с 30-дневным периодом хранения истории и с 90-дневным периодом хранения динамики изменений.

Эти элементы данных можно использовать для создания триггеров, а также при настройке условий оповещения.

Пример 1

Для создания триггера «Ошибка в веб-сценарии» вы можете указать следующее выражение триггера:

```
last(/  _  /web.test.fail[    ])<>0
```

Не забудьте заменить «Сценарий» на реальное название вашего сценария.

Пример 2

Для создания триггера «Веб-сценарий завершился с ошибкой» с полезным описанием проблемы в имени триггера вы можете указать триггер с именем:

```
"Scenario"              : {ITEM.VALUE}
```

и выражением триггера:

```
length(last(/  _  /web.test.error[    ]))>0 and last(/  _  /web.test.fail[    ])>0
```

Не забудьте заменить «Сценарий» на реальное название вашего сценария.

Пример 3

Для создания триггера «Веб-приложение медленно отвечает» вы можете указать следующее выражение триггера:

```
last(/  _  /web.test.in[    ,,bps])<10000
```

Не забудьте заменить «Сценарий» на реальное название вашего сценария.

Элементы данных шага сценария

Как только шаг создан, Zabbix автоматически добавит сопутствующие элементы данных для мониторинга.

Элемент данных	Описание
Download speed for step <Step> of scenario <Scenario> (*Скорость загрузки для шага <Шаг> сценария <Сценарий>*)	Этот элемент данных собирает информацию о скорости загрузки (байт в секунду) для указанного шага. Ключ элемента данных: web.test.in[Сценарий,Шаг,bps] Тип: *Числовой (с плавающей точкой)*
Response time for step <Step> of scenario <Scenario> (*Время ответа для шага <Шаг> сценария <Сценарий>*)	Этот элемент данных собирает информацию о времени ответа в секундах для указанного шага. Время ответа считается от начала отправки запроса до окончания получения всей передаваемой информации. Ключ элемента данных: web.test.time[Сценарий,Шаг,resp] Тип: *Числовой (с плавающей точкой)*
Response code for step <Step> of scenario <Scenario> (*Код ответа для шага <Шаг> сценария <Сценарий>*)	Этот элемент данных собирает информацию о возвращенных кодах ответа указанного шага. Ключ элемента данных: web.test.rspcode[Сценарий,Шаг] Тип: *Числовой (целое положительное)*

Будут использоваться настоящие имена сценария и шага вместо «Сценарий» и «Шаг» соответственно.

Note:
Элементы данных веб-мониторинга добавляются с 30-дневным сроком хранения истории и 90-дневным сроком хранения динамики изменений.

Note:
Если имя сценария начинается с двойных кавычек или содержит запятые либо квадратные скобки, они будут корректно экранированы в ключах элементов данных. В других случаях дополнительное экранирование не выполняется.

Эти элементы данных можно использовать для создания триггеров и настройки условий оповещения. Например, для создания триггера «Вход в веб-интерфейс Zabbix слишком медленный» вы можете указать следующее выражение триггера:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

Реальный сценарий

Обзор

В этом разделе приведен пошаговый практический пример того, как можно использовать веб-мониторинг.

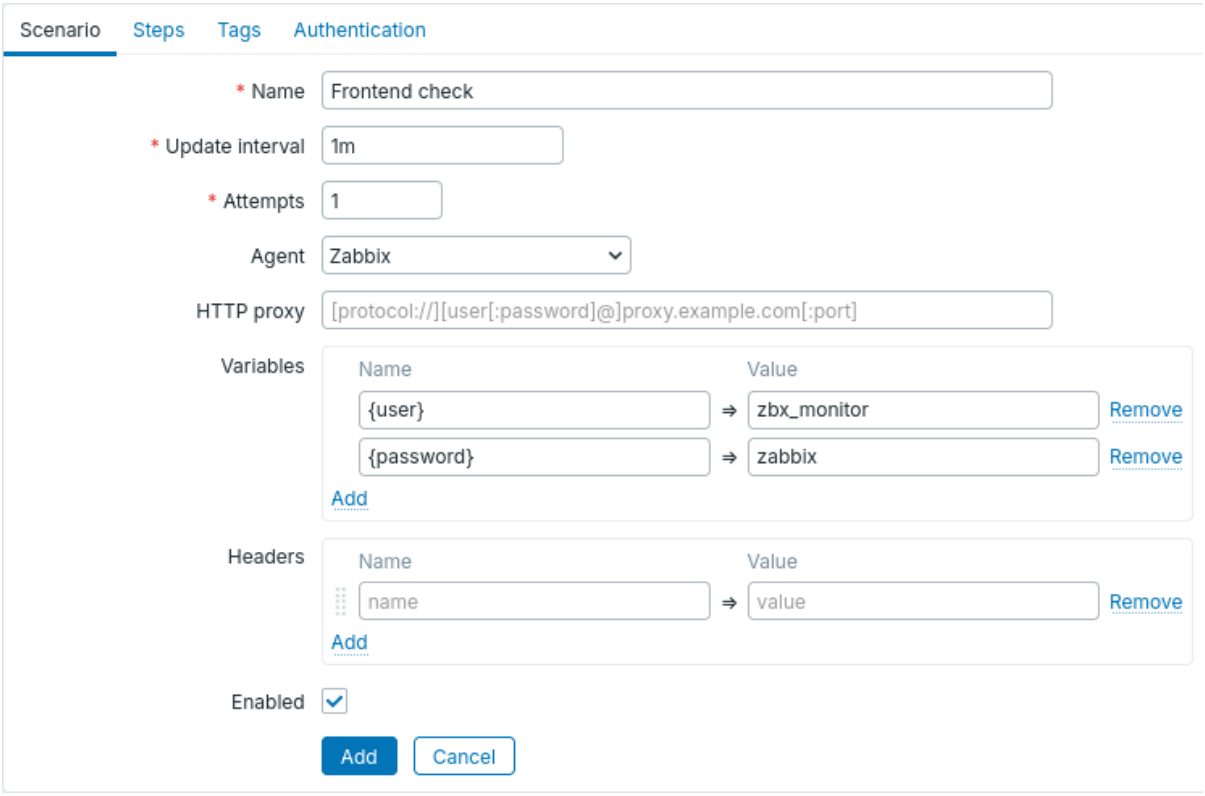
Веб-мониторинг Zabbix будет использоваться для мониторинга веб-интерфейса Zabbix. Цель состоит в том, чтобы определить, доступен ли он, предоставляет ли правильное содержимое и насколько быстро работает. Для этого

требуется выполнить несколько шагов, включая проверку доступности первой страницы, вход в систему с именем пользователя и паролем, подтверждение успешного входа, выход из системы и подтверждение выхода.

Сценарий

Добавление нового веб-сценария

1. Перейдите в *Сбор данных* > *Узлы сети*.
2. Выберите узел сети и нажмите *Web* в строке этого узла сети.
3. Нажмите *Создать веб-сценарий*.



4. В форме нового сценария на вкладке *Сценарий* заполните следующие поля:
    - *Имя*: Frontend check.
    - *Интервал обновления*: 1m.
    - *Попытки*: 1.
    - *Агент*: Zabbix.
 В разделе *Переменные* добавьте следующие пары имя-значение:
    - `{user}`: ваши существующие учетные данные пользователя Zabbix;
    - `{password}`: ваши существующие учетные данные пользователя Zabbix.

Note:
В целях безопасности рекомендуется создать отдельного пользователя с минимальными правами для использования в мониторинге.

При необходимости перейдите на вкладку *Теги* и добавьте теги веб-сценария. Теги веб-сценария можно использовать для быстрого определения связанных элементов данных и триггеров или для поиска по собранным данным. Например, подходящими тегами для этого руководства будут `component: web-scenario` и/или `target: frontend`.

Настройка шагов веб-сценария

Перейдите на вкладку *Steps* и задайте шаги для сценария.
 После настройки каждого шага нажмите кнопку *Add*.

Attention:
Обратите внимание, что веб-интерфейс Zabbix использует перенаправление JavaScript при входе в систему, поэтому вход должен выполняться первым, а функции, доступные только после входа, можно проверять только на последующих шагах. Кроме того, шаг входа должен использовать полный URL к файлу **index.php**.

Общие поля для всех шагов

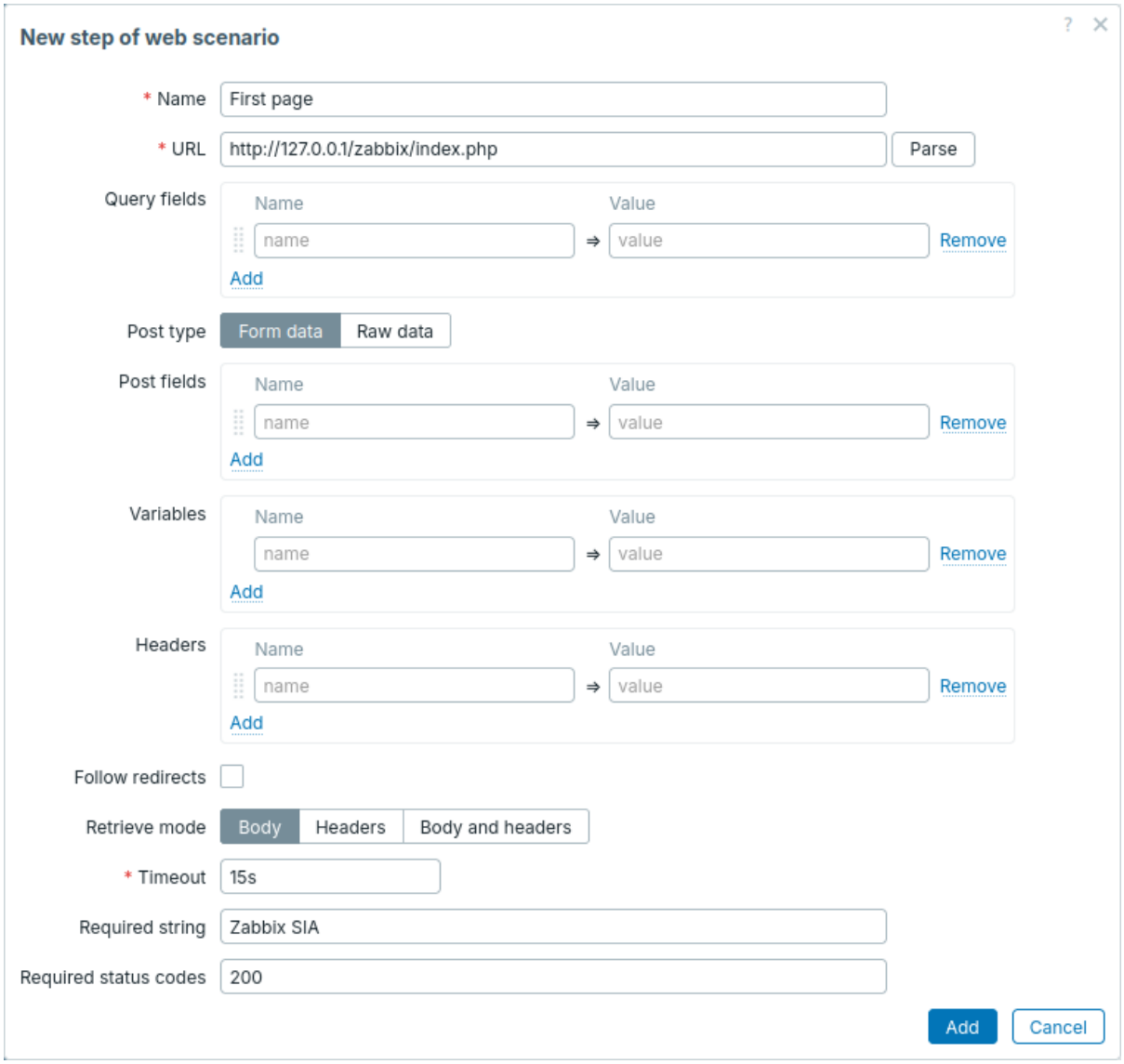
Помимо полей, специфичных для шага, заполните следующие общие поля для каждого описанного ниже шага:

- *URL*: URL веб-интерфейса Zabbix (`http://<server_ip_or_name>/zabbix/index.php` (для Apache) или `http://<server_ip_` (для NGINX)). Для шага 3 веб-сценария используйте другой URL, как описано ниже.
- *Timeout*: 15s.
- *Required status codes*: 200.

Шаг 1 веб-сценария

Проверьте, что первая страница отвечает корректно, возвращает код состояния HTTP 200 и содержит текст *Zabbix SIA*.
 Заполните следующие поля:

- *Имя*: Первая страница.
- *Тайм-аут*: 15s.
- *Обязательная строка*: Zabbix SIA.
- Общие поля для всех шагов.

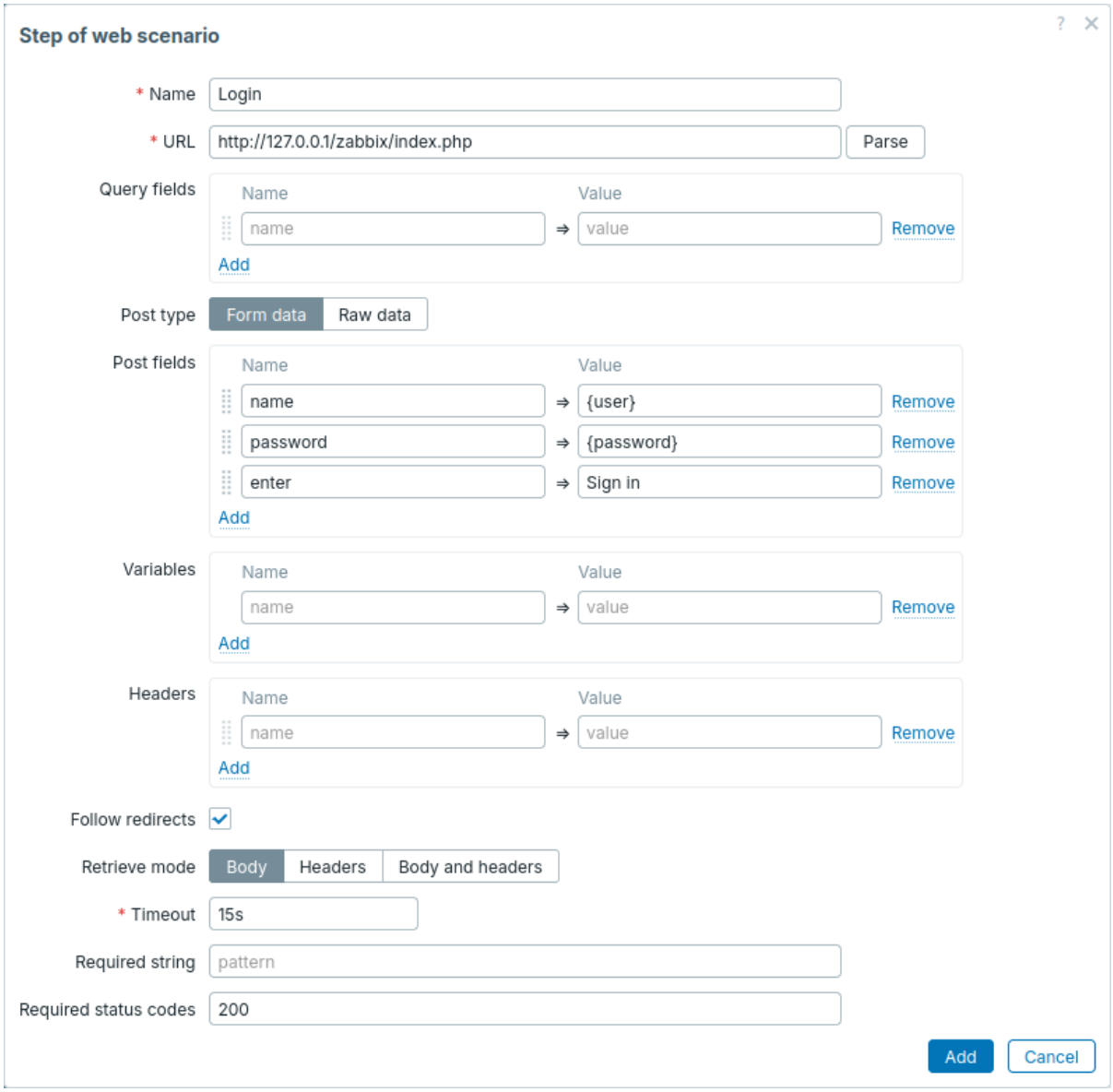


Шаг 2 веб-сценария

Войдите в веб-интерфейс Zabbix, используя макросы (переменные), определенные на вкладке *Сценарий*: *{user}* и *{password}*.
 Заполните следующие поля:

- *Name*: Login.
- В разделе *Post fields* добавьте следующие пары имя-значение:
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- Общие поля для всех шагов.

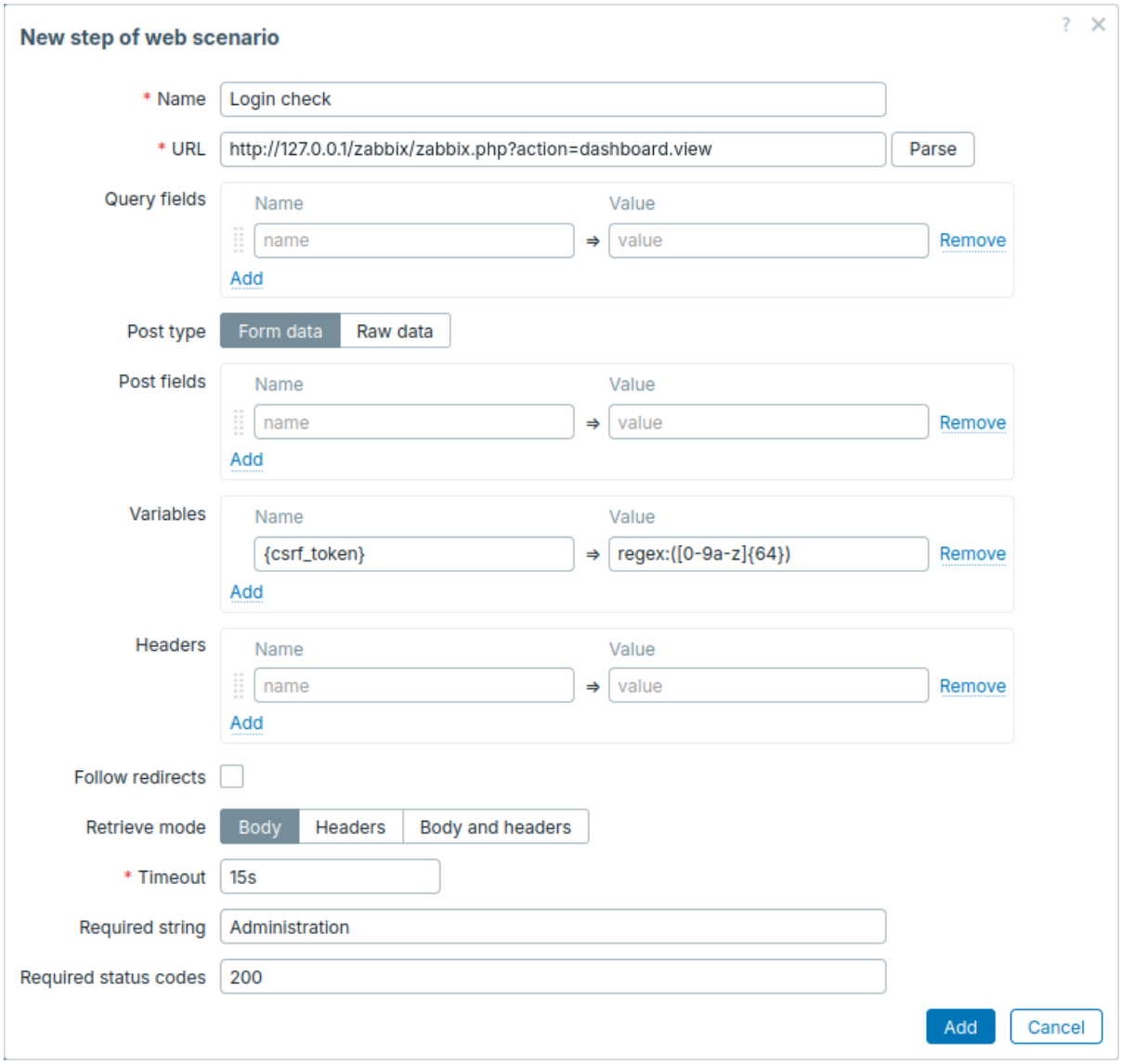
Установите флажок *Follow redirects*.



Шаг 3 веб-сценария

После входа в систему проверьте успешность, выполнив поиск строки, видимой только после входа, например, *Administration*.
 Заполните следующие поля:

- *Name*: Login check.
- *URL*: URL, на который пользователь перенаправляется после входа в систему (например, http://127.0.0.1/zabbix/zabbix.php?actio
- В разделе *Variables* добавьте пару имя-значение `{csrf_token}`: `regex:([0-9a-z]{64})`. Эта переменная будет перехватывать значение назначенного CSRF-токена для повторного использования в шаге 4 веб-сценария.
- *Required string*: Administration.
- Общие поля для всех шагов.



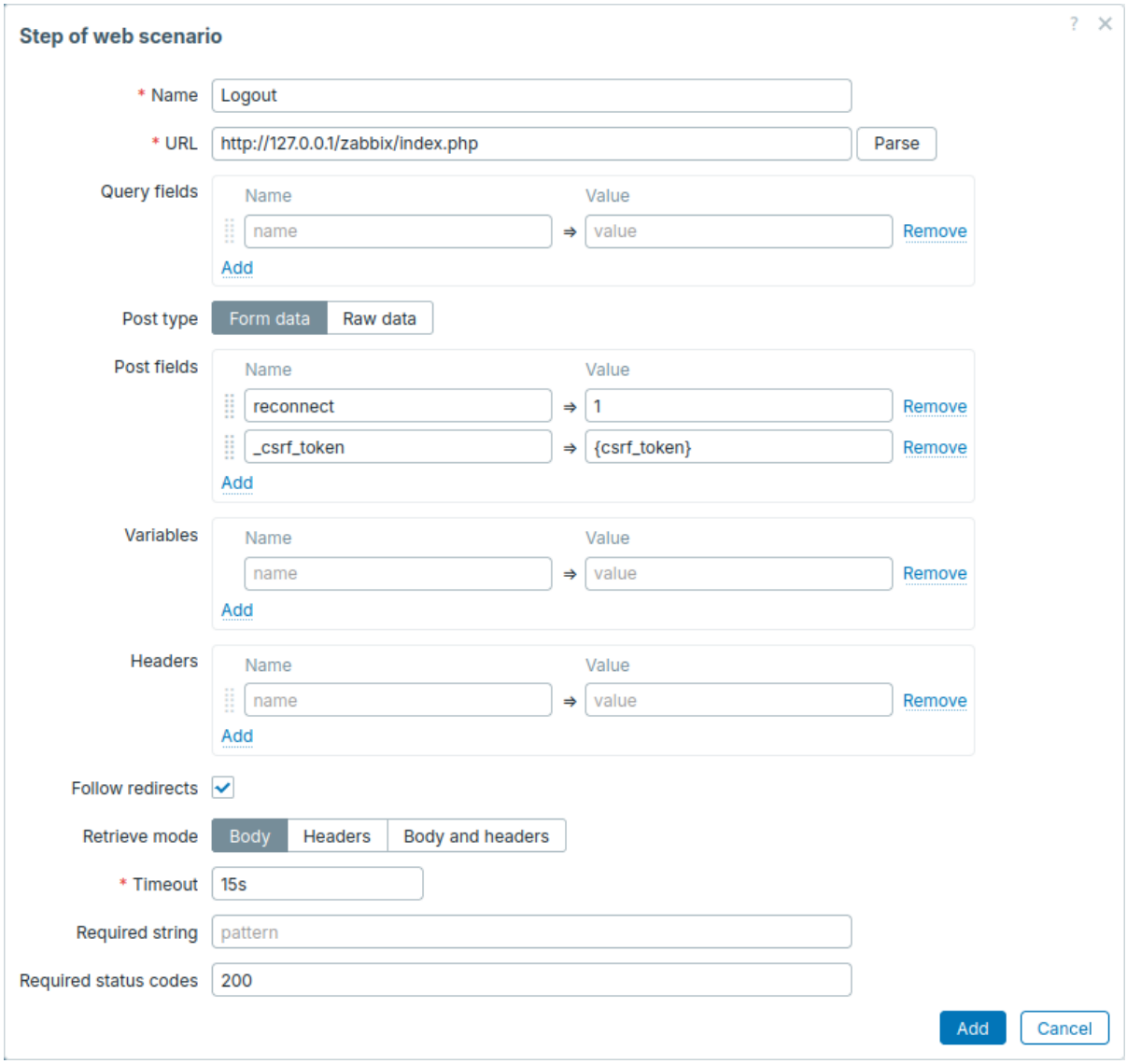
Шаг 4 веб-сценария

После проверки доступности веб-интерфейса и входа в систему добавьте шаг выхода из системы; в противном случае база данных Zabbix будет заполнена множеством открытых записей сеансов.
 Заполните следующие поля:

- *Name*: Logout.
- В разделе *Post fields* добавьте следующие пары «имя — значение»:
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`.
- Общие поля для всех шагов.

Установите флажок *Follow redirects*.

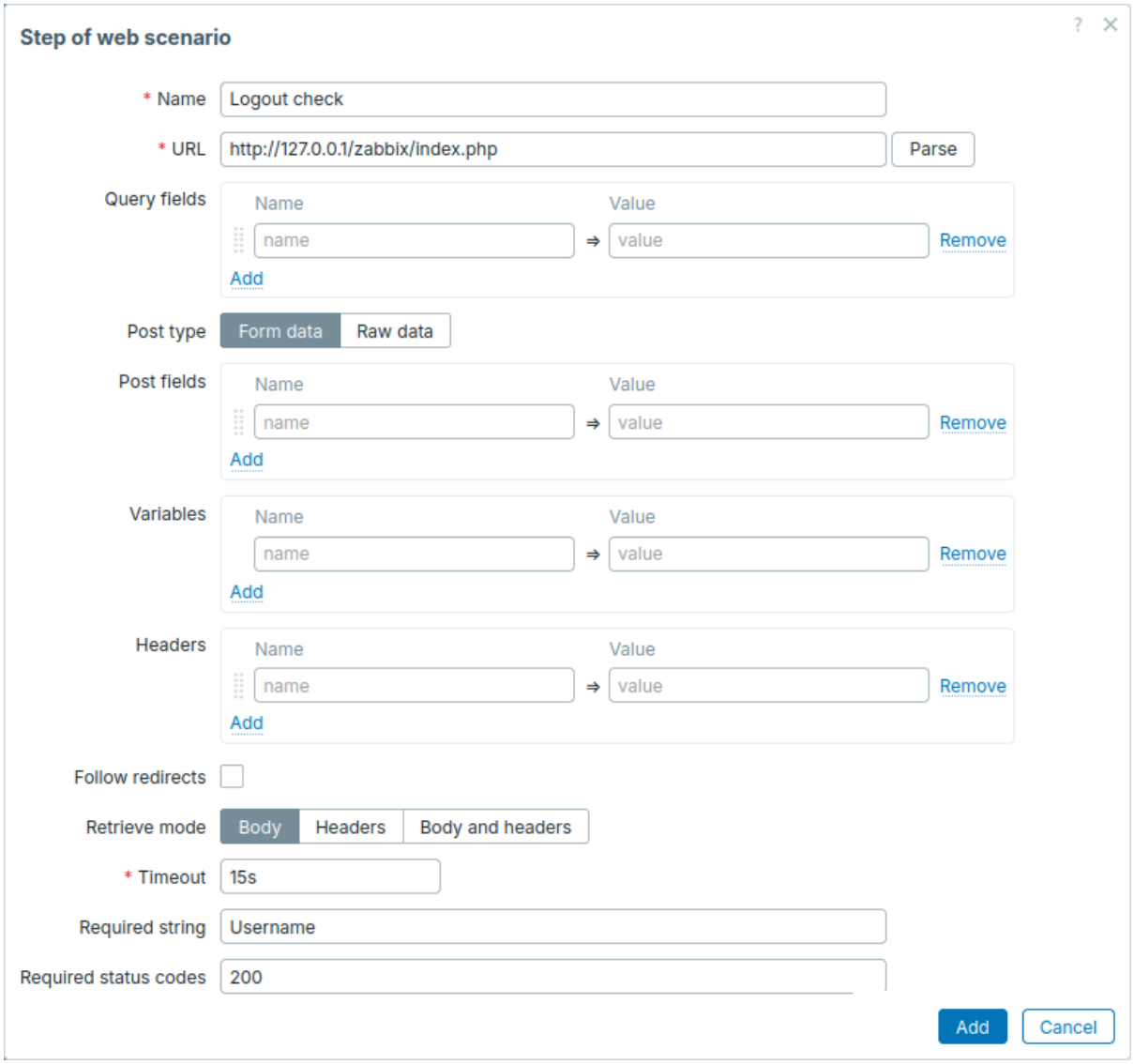
В этом шаге используется переменная `{csrf_token}`, полученная на шаге 3 веб-сценария.



Шаг 5 веб-сценария

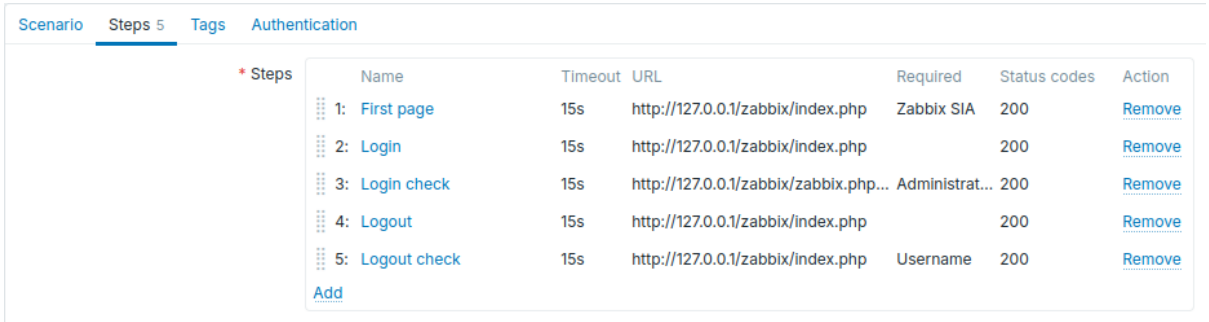
Чтобы подтвердить выход из системы, проверьте наличие строки *Username*.
 Заполните следующие поля:

- *Name*: Logout check.
- *Required string*: Username.
- Общие поля для всех шагов.



Полная конфигурация шагов

Полная конфигурация шагов веб-сценария должна выглядеть так:

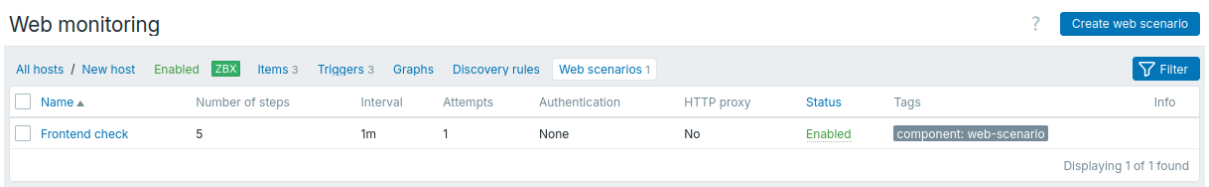


Проверьте результаты

Сохраните завершенный сценарий веб-мониторинга.

Когда веб-сценарий создан, для мониторинга автоматически добавляются новые элементы данных веб-мониторинга.

Сценарий будет добавлен к узлу сети. Чтобы просмотреть информацию о веб-сценарии, перейдите в *Monitoring* > *Hosts*, найдите узел сети в списке и нажмите гиперссылку *Web* в последнем столбце.



Нажмите на имя сценария, чтобы увидеть подробную статистику:



Мониторинг сервисов

Обзор Мониторинг сервисов — это мониторинг на уровне бизнеса, который можно использовать для получения общего представления обо всем дереве сервисов ИТ-инфраструктуры, выявления слабых мест инфраструктуры, расчета SLA различных ИТ-сервисов и просмотра другой информации на более высоком уровне. Мониторинг сервисов сосредоточен на общей доступности сервиса, а не на низкоуровневых деталях, таких как нехватка дискового пространства, высокая загрузка процессора и т. д. Мониторинг сервисов также предоставляет возможность определить первопричину проблемы, если сервис работает не так, как ожидается.

Мониторинг сервисов позволяет создавать иерархическое представление отслеживаемых данных.

Очень простая структура сервисов может выглядеть следующим образом:

```
|
|-
|  |
|  |-          1
|  |
|  |-          2
|
|-
```

Каждый узел структуры имеет атрибут состояния. Состояние рассчитывается и передается на верхние уровни в соответствии с выбранным алгоритмом. На состояние отдельных узлов влияет состояние сопоставленных проблем. Сопоставление проблем выполняется с помощью тегов.

Zabbix может отправлять уведомления или автоматически выполнять скрипт на сервере Zabbix при обнаружении изменения состояния сервиса. Можно определить гибкие правила, согласно которым родительский сервис должен перейти в «состояние проблемы» на основании состояний дочерних сервисов. Данные о проблемах сервисов можно использовать для расчета SLA и отправки отчетов SLA на основании гибкого набора условий.

Мониторинг сервисов настраивается в меню «Сервисы», которое состоит из следующих разделов:

- Сервисы

Раздел «Сервисы» позволяет создать иерархию отслеживаемой инфраструктуры, добавляя родительские сервисы, а затем дочерние сервисы к родительским сервисам.

Помимо настройки дерева сервисов, этот раздел предоставляет обзор всей инфраструктуры и позволяет быстро выявить проблемы, которые привели к изменению состояния сервиса.

- SLA

В этом разделе можно определить соглашения об уровне обслуживания и задать цели уровня обслуживания для определенных сервисов.

- Отчет SLA

В этом разделе можно просматривать отчеты SLA.

Действия сервисов

Также можно настроить действия сервисов.

Действия сервисов являются необязательными и позволяют:

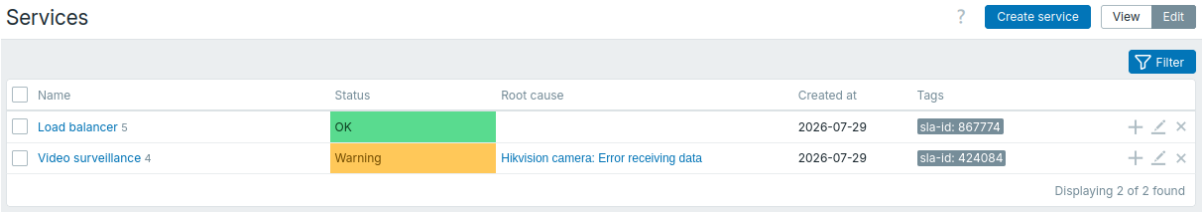
- отправлять уведомление о недоступности сервиса
- выполнять удаленную команду на сервере Zabbix при изменении состояния сервиса
- отправлять уведомление о восстановлении, когда сервис снова становится доступным.

См. также:

- пример настройки мониторинга SLA
- Примечания об обновлении сервисов из версий Zabbix ниже 6.0

Дерево услуг

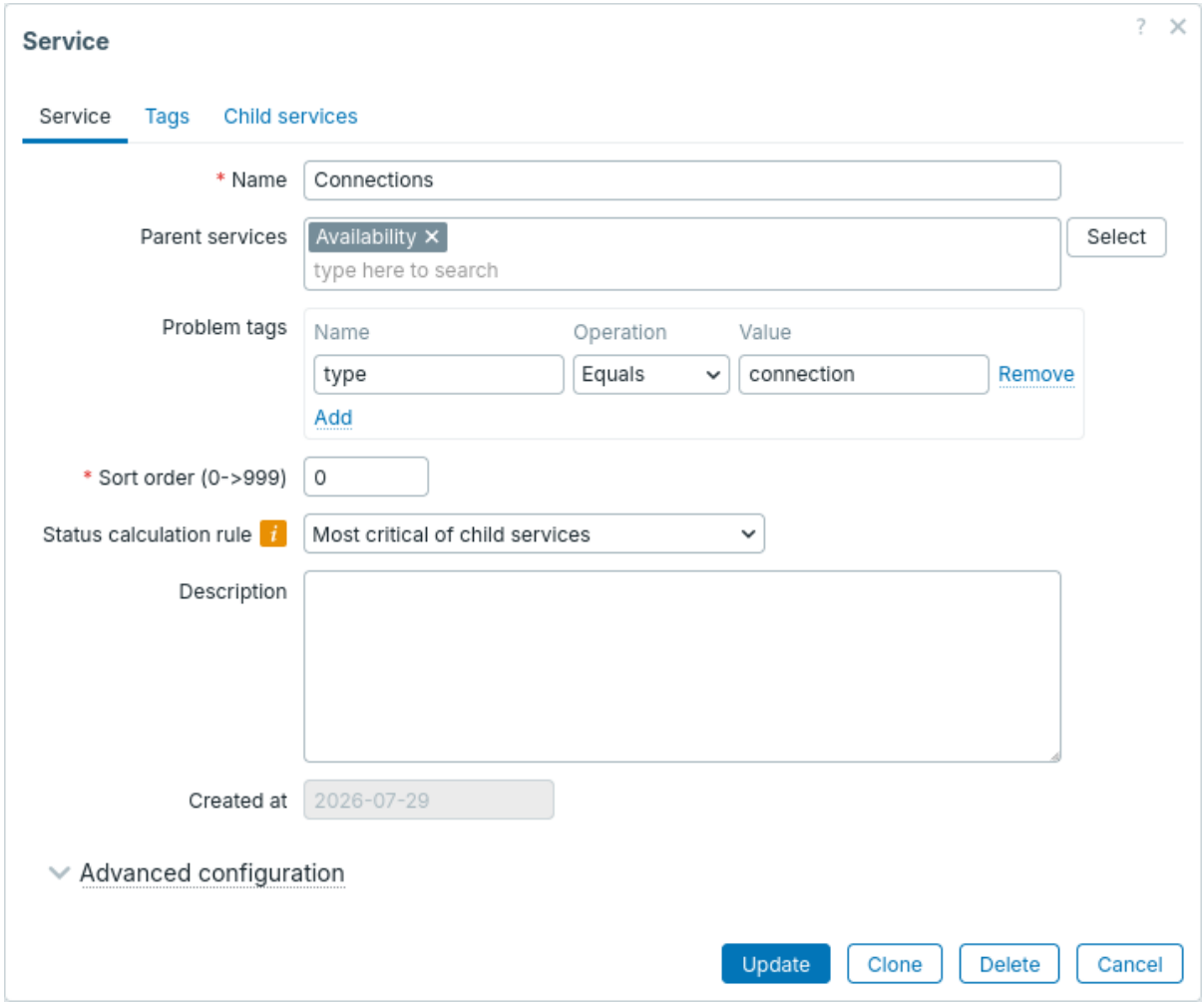
Дерево услуг настраивается в разделе меню *Services* > *Services*. В правом верхнем углу переключитесь из режима просмотра в режим редактирования.



Чтобы настроить новую услугу, нажмите кнопку *Create service* в правом верхнем углу.

Чтобы быстро добавить дочернюю услугу, можно также нажать значок плюса рядом с родительской услугой. Это откроет ту же форму настройки услуги, но параметр *Parent services* будет предварительно заполнен.

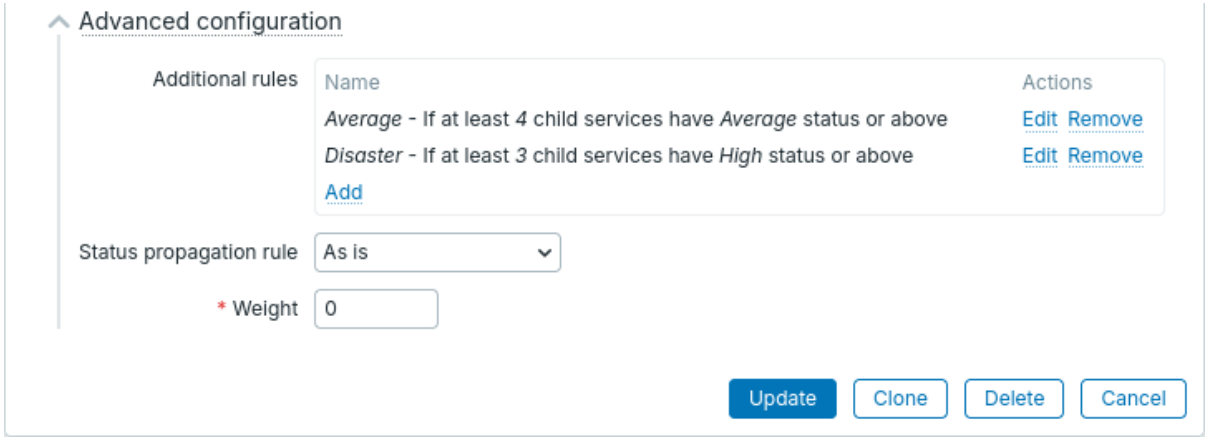
Конфигурация сервиса На вкладке **Сервис** укажите необходимые параметры сервиса:



Все обязательные поля ввода отмечены красной звездочкой.

Parameter	Description
Name	Имя сервиса.
Parent services	Родительские сервисы, к которым принадлежит сервис. Оставьте это поле пустым, если вы добавляете сервис самого верхнего уровня. Один сервис может иметь несколько родительских сервисов. В этом случае он будет отображаться в дереве сервисов под каждым из родительских сервисов.
Problem tags	Укажите теги для сопоставления данных о проблемах с сервисом. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны два оператора: **Equals** - включать указанные имена и значения тегов (с учетом регистра) **Contains** - включать указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, с учетом регистра)
Sort order	Порядок сортировки для отображения, меньшие значения идут первыми.
Status calculation rule	Правило вычисления статуса сервиса: **Most critical if all children have problems** - наиболее критичная проблема в дочерних сервисах используется для окраски статуса сервиса, если у всех дочерних сервисов есть проблемы **Most critical of child services** - наиболее критичная проблема в дочерних сервисах используется для окраски статуса сервиса **Set status to OK** - не вычислять статус сервиса Дополнительные правила вычисления статуса можно настроить в параметрах расширенной конфигурации.
Description	Описание сервиса.
Created at	Время создания сервиса; отображается при редактировании существующего сервиса.
Advanced configuration	Нажмите заголовок *Advanced configuration*, чтобы отобразить параметры расширенной конфигурации.

Расширенная настройка



Параметр	Описание
Дополнительные правила	Нажмите *Добавить*, чтобы настроить дополнительные правила расчета статуса.
Установить статус в	Установите статус сервиса в *OK* (по умолчанию), *Not classified*, *Information*, *Warning*, *Average*, *High* или *Disaster* в случае совпадения условия.
Условие	Выберите условие для прямых дочерних сервисов: **если как минимум (N) дочерних сервисов имеют статус (Status) или выше** **если как минимум (N%) дочерних сервисов имеют статус (Status) или выше** **если менее (N) дочерних сервисов имеют статус (Status) или ниже** **если менее (N%) дочерних сервисов имеют статус (Status) или ниже** **если вес дочерних сервисов со статусом (Status) или выше составляет как минимум (W)** **если вес дочерних сервисов со статусом (Status) или выше составляет как минимум (N%)** **если вес дочерних сервисов со статусом (Status) или ниже меньше (W)** **если вес дочерних сервисов со статусом (Status) или ниже меньше (N%)** Если указано несколько условий и ситуация соответствует более чем одному условию, будет установлен наивысший уровень серьезности.
N (W)	Укажите значение N или W (1-100000), либо N% (1-100) в условии.
Status	Выберите значение *Status* в условии: *OK* (по умолчанию), *Not classified*, *Information*, *Warning*, *Average*, *High* или *Disaster*.
Правило распространения статуса	Правило распространения статуса сервиса к родительскому сервису: **Как есть** - статус распространяется без изменений **Увеличить на** - можно увеличить распространяемый статус на 1-5 уровней серьезности **Уменьшить на** - можно уменьшить распространяемый статус на 1-5 уровней серьезности **Игнорировать этот сервис** - статус вообще не распространяется к родительскому сервису **Фиксированный статус** - статус распространяется статически, то есть всегда остается одинаковым
Вес	Вес сервиса (целое число в диапазоне от 0 (по умолчанию) до 1000000).

Note:
Дополнительные правила расчета статуса можно использовать только для повышения уровня серьезности по сравнению с уровнем, рассчитанным в соответствии с основным параметром *Status calculation rule*. Если согласно дополнительным правилам статус должен быть Warning, но согласно *Status calculation rule* статус - Disaster, сервис будет иметь статус Disaster.

Вкладка *Теги* содержит теги уровня сервиса. Теги уровня сервиса используются для идентификации сервиса. Теги этого типа не используются для сопоставления проблем с сервисом (для этого используйте *Теги проблем* на первой вкладке).

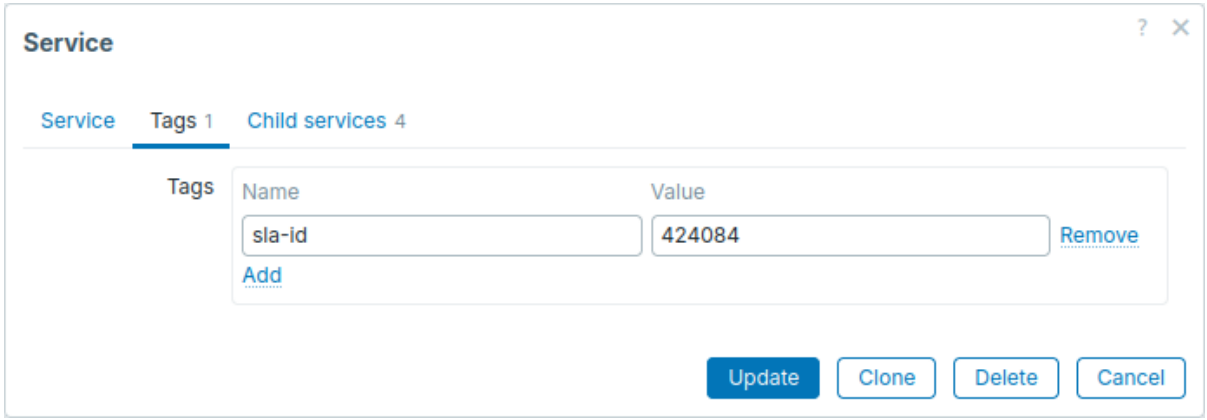
На вкладке *Дочерние сервисы* можно указать зависимые сервисы. Нажмите *Добавить*, чтобы добавить сервис из списка существующих сервисов. Если вы хотите добавить новый дочерний сервис, сначала сохраните этот сервис, затем нажмите значок плюса рядом с только что созданным сервисом.

Теги В сервисах используются два разных типа тегов:

- Теги сервиса
- Теги проблемы

Теги сервиса

Теги сервиса используются для сопоставления сервисов с действиями сервиса и SLA. Эти теги задаются на вкладке настройки сервиса *Tags*. Для сопоставления SLA используется логика *OR*: сервис будет сопоставлен с SLA, если у него есть хотя бы один совпадающий тег. В действиях сервиса правила сопоставления настраиваются и могут использовать логику *AND*, *OR* или *AND/OR*.

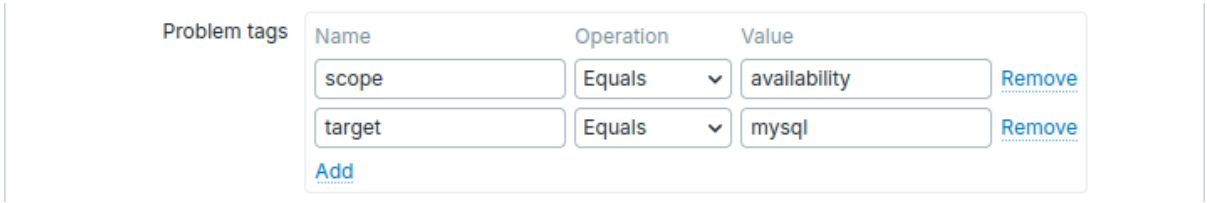


Теги проблем

Теги проблем используются для сопоставления проблем и услуг. Эти теги указываются на основной вкладке настройки услуги.

Теги проблем могут быть определены только у дочерних услуг самого нижнего уровня иерархии, и только они могут напрямую сопоставляться с проблемами. Если теги проблем совпадают, статус услуги изменится на тот же статус, что и у проблемы. В случае нескольких проблем услуга будет иметь статус самой серьезной из них. Затем статус родительской услуги вычисляется на основе статусов дочерних услуг в соответствии с правилами расчета статуса.

Если указано несколько тегов, используется логика *AND*: у проблемы должны присутствовать все теги, указанные в настройках услуги, чтобы она была сопоставлена с этой услугой.

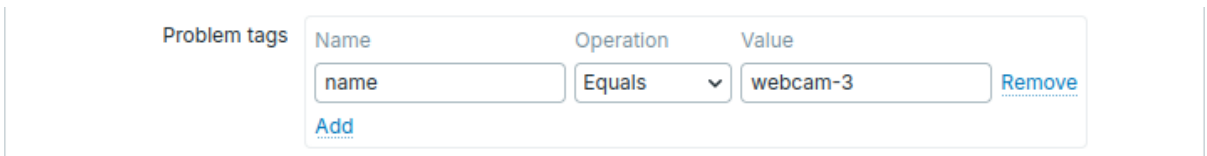


Note:
Проблема в Zabbix наследует теги по всей цепочке шаблонов, узлов сети, элементов данных, веб-сценариев и триггеров. Любой из этих тегов можно использовать для сопоставления проблем с услугами.

Пример:

Проблема *Web camera 3 is down* имеет теги `type:video-surveillance`, `floor:1` и `name:webcam-3`, а также статус *Warning*

Сервис **Web camera 3** имеет только указанный тег проблемы: `name:webcam-3`



Статус сервиса изменится с *OK* на *Warning*, когда будет обнаружена эта проблема.

Если бы у сервиса **Web camera 3** были теги проблемы `name:webcam-3` и `floor:2`, его статус не изменился бы при обнаружении проблемы, поскольку условия выполнены только частично.

Изменение существующих сервисов

> **Note:**
> Кнопки, описанные ниже, видны только тогда, когда раздел *Services* находится в реж

Чтобы изменить существующий сервис, нажмите значок карандаша рядом с сервисом.

Чтобы клонировать существующий сервис, нажмите значок карандаша, чтобы открыть его конфигурацию, а затем нажмите кнопку *Clone*. При клонировании сервиса его связи с родительскими сервисами сохраняются, а связи с дочерними сервисами — нет.

Чтобы удалить сервис, нажмите значок x рядом с ним. При удалении родительского сервиса его дочерние сервисы не будут удалены и перейдут на один уровень выше в дереве сервисов (дочерние сервисы 1-го уровня получат тот же уровень, что и удаленный родительский сервис).

Две кнопки под списком сервисов предлагают некоторые варианты массового редактирования:

- *Mass update* - массовое обновление свойств сервиса
- *Delete* - удаление сервисов

Чтобы использовать эти параметры, отметьте флажки перед соответствующими сервисами, затем нажмите нужную кнопку.

SLA

Обзор После создания сервисов можно начать отслеживать, соответствует ли их производительность соглашению об уровне обслуживания (SLA).

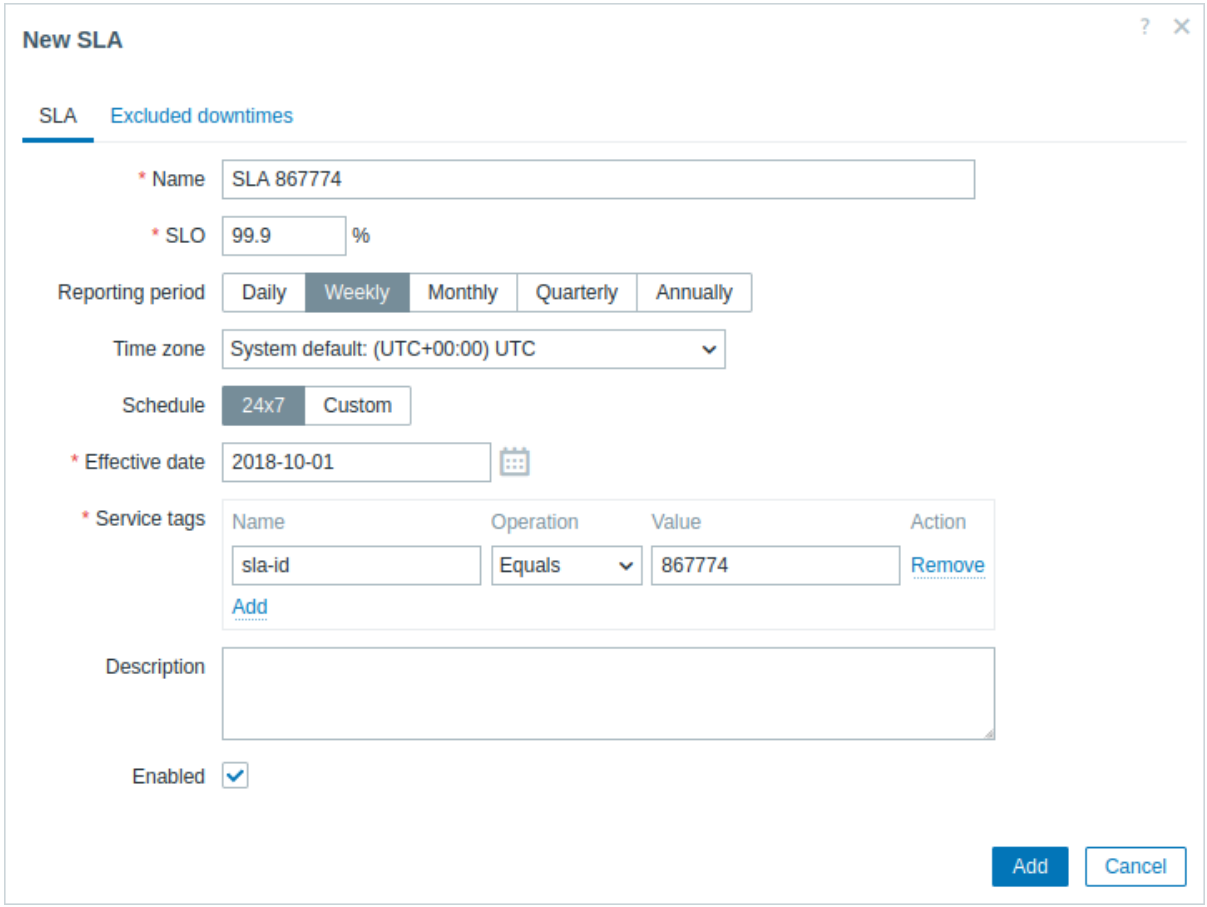
Раздел меню *Сервисы* > *SLA* позволяет настроить SLA для различных сервисов. SLA в Zabbix определяет целевой показатель уровня обслуживания (SLO), ожидаемый график доступности и запланированные периоды простоя.

SLA и сервисы сопоставляются с помощью тегов сервисов. Один и тот же SLA можно применить к нескольким сервисам — производительность будет измеряться отдельно для каждого подходящего сервиса. Одному сервису можно назначить несколько SLA — данные для каждого SLA будут отображаться отдельно.

В отчетах SLA Zabbix предоставляет данные индикатора уровня обслуживания (SLI), который измеряет фактическую доступность сервиса. Соответствие сервиса целевым показателям SLA определяется путем сравнения SLO (ожидаемой доступности в %) с SLI (фактической доступностью в %).

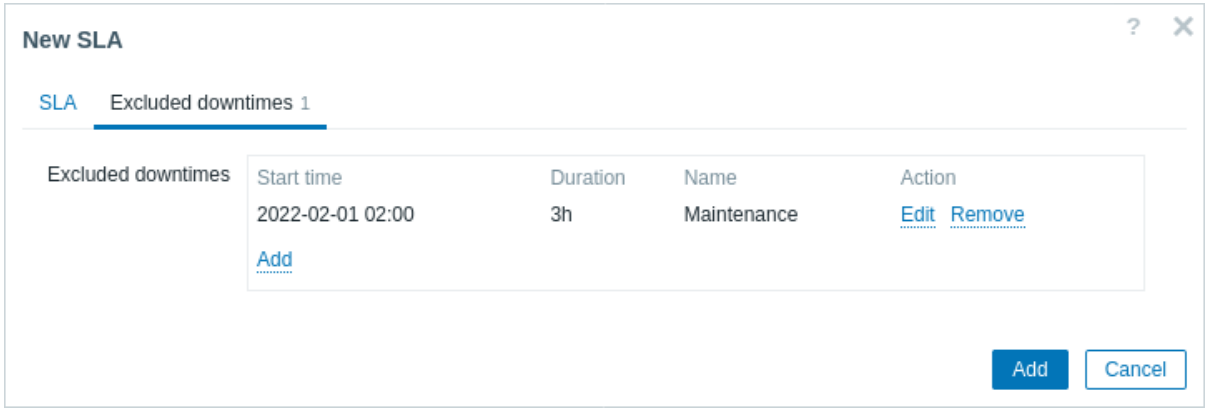
Настройка Чтобы создать новый SLA, нажмите кнопку *Создать SLA*.

На вкладке *SLA* можно указать общие параметры SLA.



Параметр	Описание
Имя	Введите имя SLA.
SLO	Введите целевой показатель уровня обслуживания (SLO) в процентах.
Отчётный период	Выберите отчётный период SLI, используемый для расчёта данных SLI в отчёте SLA: **Ежедневно** - каждый день с 00:00:00 до 23:59:59. **Еженедельно** - каждую неделю с воскресенья 00:00:00 до субботы 23:59:59. **Ежемесячно** - каждый месяц с первого дня 00:00:00 до последнего дня 23:59:59. **Ежеквартально** - каждый календарный квартал: Q1 (январь-март), Q2 (апрель-июнь), Q3 (июль-сентябрь), Q4 (октябрь-декабрь). **Ежегодно** - каждый календарный год с 1 января 00:00:00 до 31 декабря 23:59:59.
Часовой пояс	Выберите часовой пояс SLA.
Расписание	Выберите расписание SLA - 24x7 или пользовательское.
Дата начала действия	Выберите дату начала расчёта SLA.
Теги сервиса	Добавьте теги сервиса, чтобы определить сервисы, к которым следует применять этот SLA. **Имя** - имя тега сервиса; должно полностью совпадать с искомым значением с учётом регистра. **Операция** - выберите *Равно*, если значение тега должно полностью совпадать с искомым значением (с учётом регистра), или *Содержит*, если должна совпадать только часть значения тега (без учёта регистра). **Значение** - значение тега сервиса, которое необходимо найти в соответствии с выбранной операцией. SLA применяется к сервису, если совпадает хотя бы один тег сервиса.
Описание	Добавьте описание SLA.
Включён	Установите флажок, чтобы включить расчёт SLA.

На вкладке *Исключённые периоды недоступности* можно указать периоды недоступности, которые не учитываются при расчёте SLA.



Нажмите *Добавить*, чтобы настроить исключённые периоды недоступности, затем укажите название периода, дату начала и продолжительность.

Отчеты SLA Информация о том, как работает сервис по сравнению с SLA, доступна в отчете SLA. Отчеты SLA можно просматривать:

- в разделе *SLA*, щелкнув по гиперссылке отчета SLA;
- в разделе *Services*, щелкнув по имени SLA на вкладке info;
- в виджете *SLA report* на панели Dashboard.

После настройки SLA вкладка *Info* в разделе services также будет отображать некоторую информацию о производительности сервиса.

Пример настройки

Обзор В этом разделе описана простая настройка мониторинга кластера высокой доступности Zabbix как услуги.

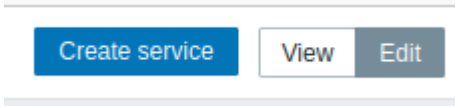
Предварительные требования Перед тем как настраивать мониторинг услуг, вам необходимо настроить следующие узлы сети:

- *HA нода 1* с по крайней мере одним триггером и тегом (желательно задать на уровне триггера) `component:ha-node-1`
- *HA нода 2* с по крайней мере одним триггером и тегом (желательно задать на уровне триггера) `component:ha-node-2`

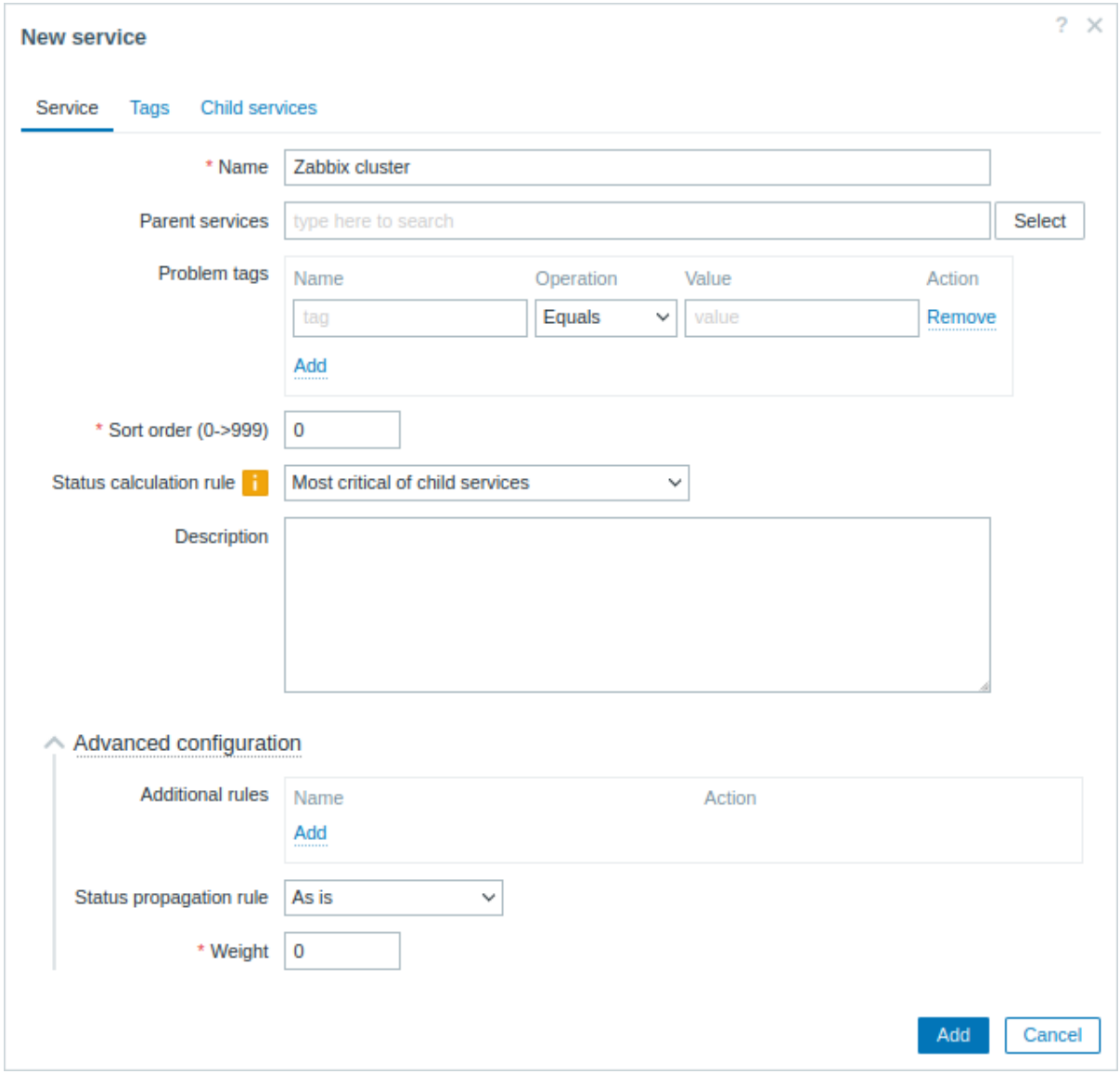
Дерево сервисов Следующий шаг — построить дерево сервисов. В этом примере инфраструктура очень простая и состоит из трех сервисов: *Zabbix cluster* (родительский) и двух дочерних сервисов *Zabbix server node 1* и *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

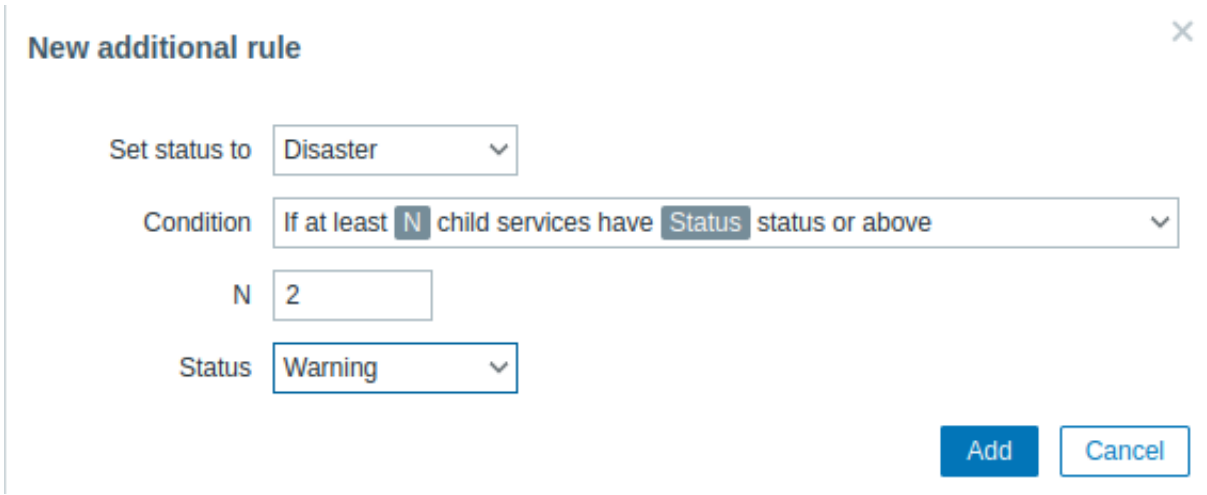
На странице *Services* включите режим *Edit* и нажмите *Create service*:



В окне настройки сервиса введите имя *Zabbix cluster* и нажмите заголовок *Advanced configuration*, чтобы отобразить дополнительные параметры настройки.



Настройте дополнительное правило:

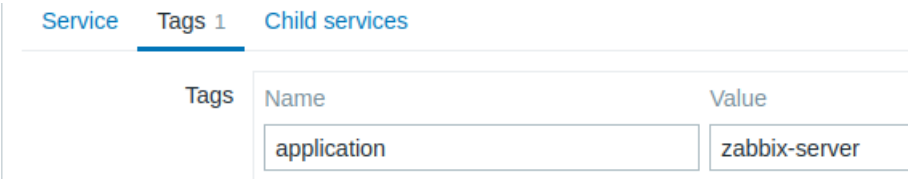


У *Zabbix cluster* будет два дочерних сервиса — по одному для каждого из узлов HA. Если у обоих узлов HA возникнут проблемы со статусом не ниже *Warning*, статус родительского сервиса должен быть установлен в *Disaster*. Чтобы этого добиться, дополнительное правило следует настроить так:

- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2
- Status: Warning

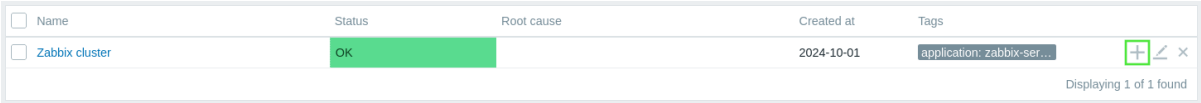
Перейдите на вкладку *Tags* и добавьте тег `application:zabbix-server`. Этот тег будет использоваться позже для

действий с сервисами и отчетов SLA.



Сохраните новый сервис.

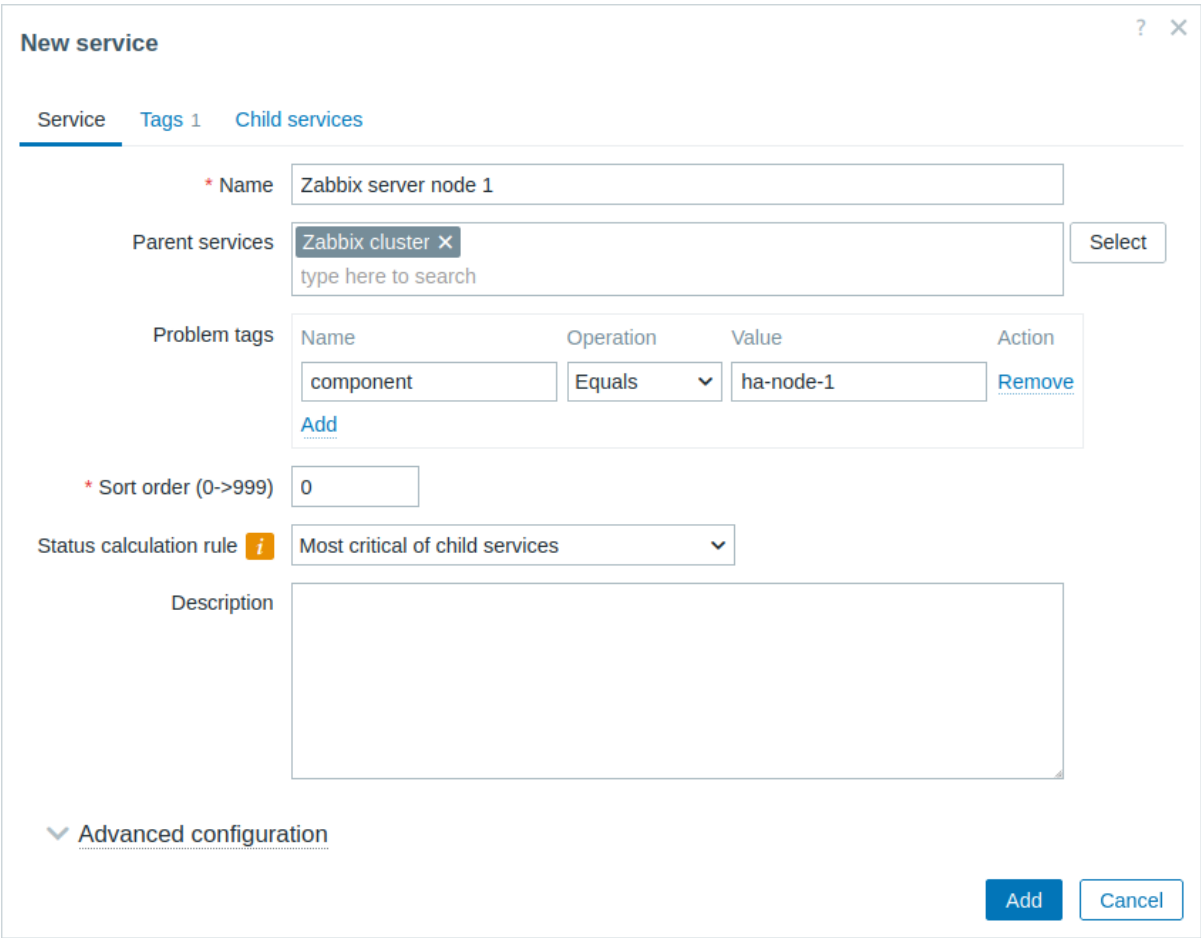
Чтобы добавить дочернюю услугу, нажмите значок плюса рядом с услугой кластера Zabbix (значок виден только в режиме *Edit*).



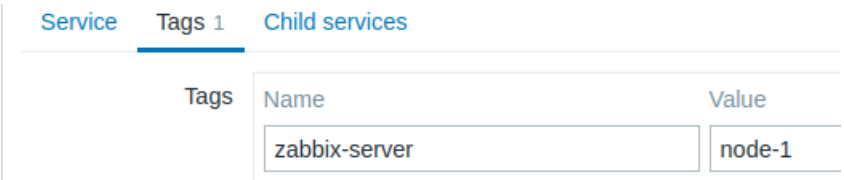
В окне настройки услуги введите имя *Zabbix server node 1*. Обратите внимание, что параметр Parent services уже предварительно заполнен значением *Zabbix cluster*.

Доступность этой услуги зависит от проблем на узле сети *HA node 1*, отмеченных тегом проблемы `component:ha-node-1`. В параметре Problem tags введите:

- Name: component
- Operation: Equals
- Value: ha-node-1



Перейдите на вкладку *Tags* и добавьте тег услуги: `zabbix-server:node-1`. Этот тег будет использоваться позже для действий с услугами и отчетов SLA.



Сохраните новую услугу.

Создайте еще одну дочернюю для Zabbix кластера услугу с именем «Zabbix сервер нода 2».

Задайте теги проблем следующим образом:

- Имя: component
- Операция: Равно
- Значение: ha-node-2

Переключитесь на вкладку *Теги* и добавьте тег услуги: `zabbix-server:node-2`.

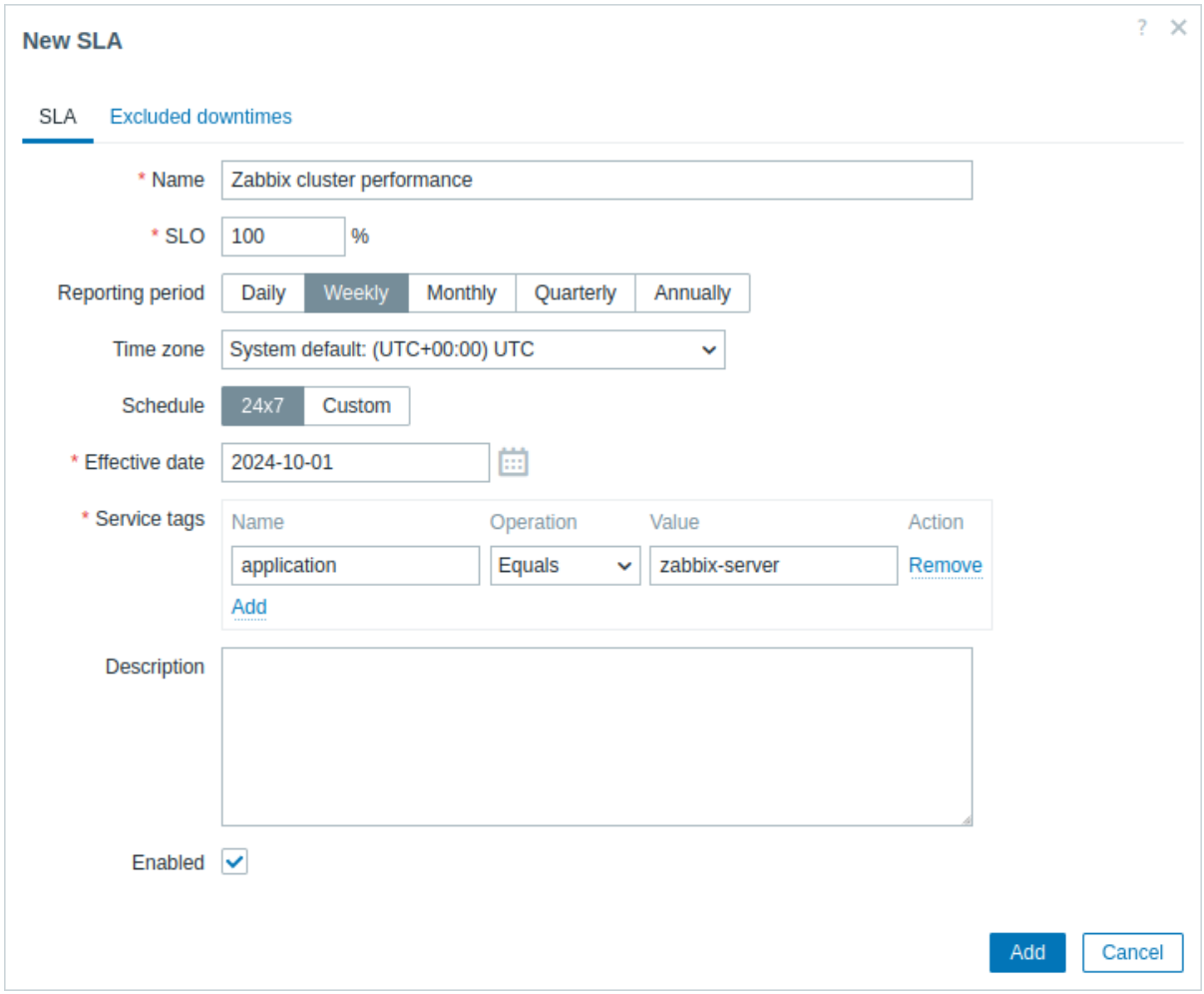
Сохраните новую услугу.

SLA В этом примере ожидаемая производительность кластера Zabbix составляет 100%, за исключением одного часа планового обслуживания два раза в год.

Сначала необходимо добавить новое соглашение об уровне обслуживания:

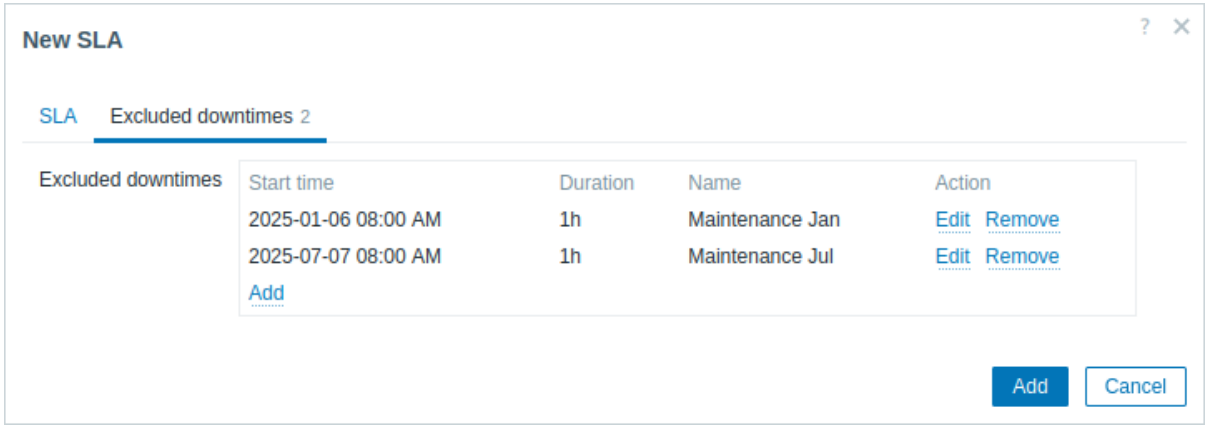
1. Перейдите в раздел *Services* > *SLA*.
2. Нажмите *Create SLA*.
3. Введите имя *Zabbix cluster performance* и установите SLO равным 100%. Сервис Zabbix cluster имеет тег сервиса `application:zabbix-server`.
4. Чтобы использовать это SLA для измерения производительности кластера Zabbix, в параметре *Service tags* укажите:

- Name: application
- Operation: Equals
- Value: zabbix-server



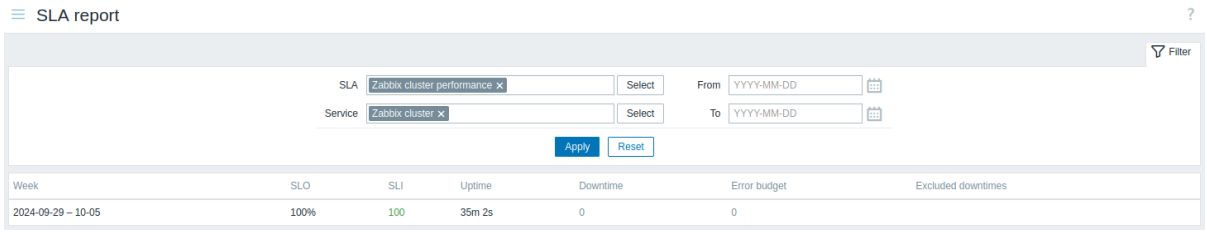
В реальной конфигурации также можно изменить требуемый отчетный период, часовой пояс и дату начала либо изменить расписание с режима 24/7 на пользовательское. Для этого примера достаточно настроек по умолчанию.

5. Перейдите на вкладку *Excluded downtimes* и добавьте периоды недоступности для запланированных периодов обслуживания, чтобы исключить их из расчета SLA.
6. В разделе *Excluded downtimes* нажмите ссылку *Add*, укажите название периода недоступности, запланированное время начала и продолжительность.

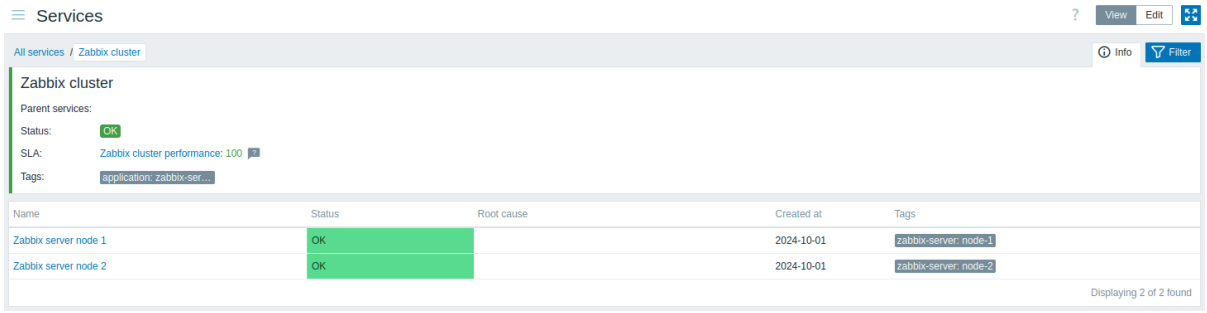


7. Нажмите *Add*, чтобы сохранить новое SLA.

Перейдите в раздел отчетов SLA, чтобы просмотреть отчет SLA для кластера Zabbix.



Информацию о SLA также можно проверить в разделе *Services*.



Обновление мониторинга сервисов

Обзор В Zabbix 6.0 функциональность мониторинга сервисов была существенно переработана (см. Что нового в Zabbix 6.0.0 для списка изменений).

На этой странице описано, как сервисы и SLA, определенные в более ранних версиях Zabbix, изменяются при обновлении до Zabbix 6.0 или более новой версии.

Службы В более старых версиях Zabbix у служб было два типа зависимостей: soft и hard. После обновления все зависимости станут равнозначными.

Если служба ”Child service” ранее была связана с ”Parent service 1” через hard dependency и дополнительно с ”Parent service 2” через soft dependency, то после обновления у службы ”Child service” будет две родительские службы: ”Parent service 1” и ”Parent service 2”.

Сопоставление между проблемами и службами на основе триггеров было заменено сопоставлением на основе тегов. В Zabbix 6.0 и новее в форме настройки службы появился новый параметр *Problem tags*, который позволяет указать одну или несколько пар имя тега и значение для сопоставления проблем. Триггеры, которые были связаны со службой, получат новый тег `ServiceLink` : `<trigger ID>:<trigger name>` (значение тега будет обрезано до 32 символов). Связанные службы получат `ServiceLink` тег проблемы с тем же значением.

Правила расчета статуса

Алгоритм расчета статуса будет обновлен с использованием следующих правил:

- Не вычислять → Установить статус в OK

- Проблема, если хотя бы один дочерний элемент имеет проблему → Наиболее критичный из дочерних сервисов
- Проблема, если все дочерние элементы имеют проблемы → Наиболее критичный, если все дочерние элементы имеют проблемы

SLA Ранее целевые показатели SLA приходилось определять отдельно для каждого сервиса. Начиная с Zabbix 6.0, SLA стал отдельной сущностью, которая содержит информацию о расписании сервиса, ожидаемом целевом уровне обслуживания (SLO) и периодах простоя, которые нужно исключать из расчета. После настройки SLA можно назначить нескольким сервисам через теги сервиса.

Во время обновления:

- Идентичные SLA, определенные для каждого сервиса, будут сгруппированы, и для каждой группы будет создан один SLA.
- Каждый затронутый сервис получит специальный тег `SLA:<ID>`, и такой же тег будет указан в параметре *Теги сервиса* соответствующего SLA.
- Время создания сервиса, новый показатель в отчетах SLA, для существующих сервисов будет установлено в 01/01/2000 00:00.

Шифрование и безопасность

Шифрование Zabbix поддерживает зашифрованные соединения между компонентами Zabbix с использованием протокола Transport Layer Security (TLS) версии 1.2 и 1.3 (в зависимости от криптографической библиотеки). Поддерживается шифрование на основе сертификатов и на основе предварительно общего ключа.

Шифрование можно настроить для соединений:

- Между сервером Zabbix, прокси Zabbix, агентом Zabbix, веб-службой Zabbix, утилитами zabbix_sender и zabbix_get
- С базой данных Zabbix из веб-интерфейса Zabbix и server/proxy
- Между веб-интерфейсом Zabbix и сервером Zabbix

Шифрование является необязательным и настраивается для отдельных компонентов:

- Некоторые прокси и агенты можно настроить на использование шифрования на основе сертификатов с сервером, тогда как другие могут использовать шифрование на основе предварительно общего ключа, а еще другие могут продолжать использовать незашифрованные соединения (как и раньше).
- Сервер (прокси) может использовать разные настройки шифрования для разных узлов сети.

Демонные программы Zabbix используют один прослушивающий порт для зашифрованных и незашифрованных входящих соединений. Добавление шифрования не требует открытия новых портов на межсетевых экранах.

Ограничения

- Закрытые ключи хранятся в виде обычного текста в файлах, доступных для чтения компонентам Zabbix при запуске.
- Предварительно общие ключи вводятся во веб-интерфейсе Zabbix и хранятся в базе данных Zabbix в виде обычного текста.
- Встроенное шифрование не защищает обмен данными между веб-сервером, на котором работает веб-интерфейс Zabbix, и веб-браузером пользователя.
- В настоящее время каждое зашифрованное соединение открывается с полным TLS-handshake, кэширование сеансов и tickets не реализованы.
- Добавление шифрования увеличивает время проверки элементов данных и выполнения действий в зависимости от сетевой задержки:
  - Например, если задержка пакета составляет 100 мс, то открытие TCP-соединения и отправка незашифрованного запроса занимает около 200 мс. При шифровании для установления TLS-соединения добавляется примерно 1000 мс.
  - Может потребоваться увеличить значения тайм-аутов, иначе некоторые элементы данных и действия, выполняющие удаленные сценарии на агентах, могут работать с незашифрованными соединениями, но завершаться по тайм-ауту при использовании шифрования.
- Шифрование не поддерживается сетевым обнаружением. Проверки агента Zabbix, выполняемые в рамках сетевого обнаружения, будут незашифрованными, и если агент Zabbix настроен на отклонение незашифрованных соединений, такие проверки не будут успешными.

Компиляция Zabbix с поддержкой шифрования Для поддержки шифрования Zabbix необходимо скомпилировать и связать с одной из поддерживаемых криптографических библиотек:

- GnuTLS - начиная с версии 3.1.18

- OpenSSL - версии 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x - 3.5.x
- LibreSSL - протестировано с версиями 2.7.4, 2.8.2:
  - LibreSSL 2.6.x не поддерживается
  - LibreSSL поддерживается как совместимая замена OpenSSL; новые специфичные для LibreSSL функции API `tls_*()` не используются. Компоненты Zabbix, скомпилированные с LibreSSL, не смогут использовать PSK, можно использовать только сертификаты.

Note:
Дополнительную информацию о настройке SSL для веб-интерфейса Zabbix можно найти в разделе рекомендаций по безопасности.

Библиотека выбирается путем указания соответствующего параметра для скрипта `configure`:

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (также используется для LibreSSL)

Например, для настройки исходного кода сервера и агента с *OpenSSL* можно использовать следующую команду:

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

Различные компоненты Zabbix можно компилировать с разными криптографическими библиотеками (например, сервер с *OpenSSL*, а агент с *GnuTLS*).

Attention:
Если вы планируете использовать предварительно установленные ключи (PSK), рассмотрите возможность использования библиотек *GnuTLS* или *OpenSSL 1.1.0* (или более новых) в компонентах Zabbix, использующих PSK. Библиотеки *GnuTLS* и *OpenSSL 1.1.0* поддерживают наборы шифров PSK с идеальной прямой секретностью. Более старые версии библиотеки *OpenSSL* (1.0.1, 1.0.2c) также поддерживают PSK, но доступные наборы шифров PSK не обеспечивают идеальную прямую секретность.

Управление шифрованием соединений В Zabbix можно использовать следующие типы соединений:

- без шифрования (по умолчанию)
- шифрование на основе сертификатов RSA
- шифрование на основе PSK

Для указания шифрования между компонентами Zabbix используются два важных параметра:

- TLSConnect - указывает, какое шифрование использовать для исходящих соединений (без шифрования, PSK или сертификат)
- TLSAccept - указывает, какие типы соединений разрешены для входящих соединений (без шифрования, PSK или сертификат). Можно указать одно или несколько значений.

`TLSConnect` используется в конфигурационных файлах прокси Zabbix (в активном режиме указывает только соединения с сервером) и агента Zabbix (для активных проверок). Эквивалентом `TLSConnect` в веб-интерфейсе Zabbix является поле *Соединения с узлом сети* на вкладке *Шифрование* в разделе *Сбор данных* > *Узлы сети* > <некоторый узел сети> и поле *Соединения с прокси* на вкладке *Шифрование* в разделе *Администрирование* > *Прокси* > <некоторый прокси>. Если настроенный тип шифрования соединения не работает, другие типы шифрования использоваться не будут.

`TLSAccept` используется в конфигурационных файлах прокси Zabbix (в пассивном режиме указывает только соединения от сервера) и агента Zabbix (для пассивных проверок). Эквивалентом TLSAccept в веб-интерфейсе Zabbix является поле *Соединения от узла сети* в разделе *Сбор данных* > *Узлы сети* > <некоторый узел сети> > вкладка *Шифрование* и поле *Соединения от прокси* в разделе *Администрирование* > *Прокси* > <некоторый прокси> > вкладка *Шифрование*.

Обычно для входящих соединений настраивается только один тип шифрования. Однако может потребоваться изменить тип шифрования, например с незашифрованного на шифрование на основе сертификата, с минимальным временем простоя и возможностью отката. Для этого:

- Установите `TLSAccept=unencrypted,cert` в конфигурационном файле агента и перезапустите агент Zabbix.
- Проверьте соединение с помощью `zabbix_get` с агентом, используя сертификат. Если соединение работает, можно изменить настройки шифрования для этого агента в веб-интерфейсе Zabbix в разделе *Сбор данных* > *Узлы сети* > <некоторый узел сети> > вкладка *Шифрование*, установив для поля *Соединения с узлом сети* значение *Сертификат*.
- После обновления кэша конфигурации сервера (и обновления конфигурации прокси, если мониторинг узла сети выполняется через прокси) соединения с этим агентом будут зашифрованы.
- Если все работает как ожидается, установите `TLSAccept=cert` в конфигурационном файле агента и перезапустите агент Zabbix. Теперь агент будет принимать только зашифрованные соединения на основе сертификатов. Незашифрованные соединения и соединения на основе PSK будут отклоняться.

Аналогичным образом это работает на сервере и прокси. Если в веб-интерфейсе Zabbix в настройках узла сети для поля *Соединения от узла сети* установлено значение «Сертификат», от агента (активные проверки) и zabbix_sender (элементы данных типа trapper) будут приниматься только зашифрованные соединения на основе сертификатов.

Скорее всего, для входящих и исходящих соединений будет настроен один и тот же тип шифрования либо полное отсутствие шифрования. Однако технически можно настроить их асимметрично, например использовать шифрование на основе сертификата для входящих соединений и на основе PSK для исходящих.

Настройки шифрования для каждого узла сети отображаются в веб-интерфейсе Zabbix в разделе *Сбор данных* > *Узлы сети*, в столбце *Шифрование агента*. Например:

Пример	Соединения с узлом сети	Разрешенные соединения от узла сети	Отклоненные соединения от узла сети
NONE	Без шифрования	Без шифрования	Зашифрованные соединения, соединения на основе сертификатов и PSK
CERT NONE PSK CERT	Зашифрованные, на основе сертификата	Зашифрованные, на основе сертификата	Незашифрованные и зашифрованные на основе PSK
PSK NONE PSK CERT	Зашифрованные, на основе PSK	Зашифрованные, на основе PSK	Незашифрованные и зашифрованные на основе сертификата
PSK NONE PSK CERT	Зашифрованные, на основе PSK	Незашифрованные и зашифрованные на основе PSK	Зашифрованные, на основе сертификата
CERT NONE PSK CERT	Зашифрованные, на основе сертификата	Незашифрованные, зашифрованные на основе PSK или сертификата	-

Attention:
По умолчанию соединения не шифруются. Шифрование необходимо настраивать отдельно для каждого узла сети и прокси.

zabbix_get и zabbix_sender с шифрованием См. страницы руководства zabbix_get и zabbix_sender для использования их с шифрованием.

Наборы шифров Наборы шифров по умолчанию настраиваются внутренне во время запуска Zabbix.

Также поддерживаются наборы шифров, настроенные пользователем, для GnuTLS и OpenSSL. Пользователи могут настроить наборы шифров в соответствии со своими политиками безопасности. Использование этой функции необязательно (встроенные наборы шифров по умолчанию по-прежнему работают).

Для криптографических библиотек, собранных с настройками по умолчанию, встроенные правила Zabbix обычно приводят к следующим наборам шифров (в порядке от более высокого к более низкому приоритету):

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA

Library	Certificate ciphersuites	PSK ciphersuites
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Пользовательские наборы шифров Встроенные критерии выбора набора шифров можно переопределить с помощью пользовательских наборов шифров.

Attention:
Пользовательские наборы шифров — это функция, предназначенная для опытных пользователей, которые понимают наборы шифров TLS, их безопасность и последствия ошибок, а также уверенно работают с устранением неполадок TLS.

Встроенные критерии выбора набора шифров можно переопределить с помощью следующих параметров:

Область переопределения	Параметр	Значение	Описание
Выбор набора шифров для сертификатов	TLSCipherCert13	Допустимые строки шифров OpenSSL 1.1.1 cipher strings для протокола TLS 1.3 (их значения передаются функции OpenSSL `SSL_CTX_set_ciphersuites()`).	Критерии выбора набора шифров на основе сертификатов для TLS 1.3 Только OpenSSL 1.1.1 или новее.
	TLSCipherCert	Допустимые строки шифров OpenSSL cipher strings для TLS 1.2 или допустимые строки приоритетов GnuTLS priority strings. Их значения передаются функциям `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()` соответственно.	Критерии выбора набора шифров на основе сертификатов для TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Выбор набора шифров для PSK	TLSCipherPSK13	Допустимые строки шифров OpenSSL 1.1.1 cipher strings для протокола TLS 1.3 (их значения передаются функции OpenSSL `SSL_CTX_set_ciphersuites()`).	Критерии выбора набора шифров на основе PSK для TLS 1.3 Только OpenSSL 1.1.1 или новее.
	TLSCipherPSK	Допустимые строки шифров OpenSSL cipher strings для TLS 1.2 или допустимые строки приоритетов GnuTLS priority strings. Их значения передаются функциям `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()` соответственно.	Критерии выбора набора шифров на основе PSK для TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Область переопределения	Параметр	Значение	Описание
Общий список наборов шифров для сертификата и PSK	TLSCipherAll13	Допустимые строки шифров OpenSSL 1.1.1 cipher strings для протокола TLS 1.3 (их значения передаются функции OpenSSL `SSL_CTX_set_ciphersuites()`).	Критерии выбора набора шифров для TLS 1.3 Только OpenSSL 1.1.1 или новее.
	TLSCipherAll	Допустимые строки шифров OpenSSL cipher strings для TLS 1.2 или допустимые строки приоритетов GnuTLS priority strings. Их значения передаются функциям `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()` соответственно.	Критерии выбора набора шифров для TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Чтобы переопределить выбор набора шифров в утилитах zabbix_get и zabbix_sender, используйте параметры командной строки:

- `--tls-cipher13`
- `--tls-cipher`

Новые параметры являются необязательными. Если параметр не указан, используется внутреннее значение по умолчанию. Если параметр задан, он не может быть пустым.

Если при задании значения TLSCipher* в криптографической библиотеке возникает ошибка, сервер, прокси или агент не запустится, и в журнал будет записана ошибка.

Важно понимать, когда применим каждый параметр.

Исходящие соединения

Самый простой случай — исходящие соединения:

- Для исходящих соединений с сертификатом используйте TLSCipherCert13 или TLSCipherCert
- Для исходящих соединений с PSK используйте TLSCipherPSK13 или TLSCipherPSK
- В случае утилит zabbix_get и zabbix_sender можно использовать параметры командной строки `--tls-cipher13` или `--tls-cipher` (шифрование однозначно задается параметром `--tls-connect`)

Входящие соединения

С входящими соединениями все немного сложнее, поскольку правила зависят от компонентов и конфигурации.

Для Zabbix **агент**:

Настройка подключения агента	Конфигурация шифров
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Для Zabbix **сервер** и **прокси**:

Настройка подключения	Конфигурация шифров
Исходящие соединения с использованием PSK	TLSCipherPSK, TLSCipherPSK13
Входящие соединения с использованием сертификатов	TLSCipherAll, TLSCipherAll13
Входящие соединения с использованием PSK, если у сервера нет сертификата	TLSCipherPSK, TLSCipherPSK13
Входящие соединения с использованием PSK, если у сервера есть сертификат	TLSCipherAll, TLSCipherAll13

В двух таблицах выше можно заметить некоторую закономерность:

- TLSCipherAll и TLSCipherAll13 можно указывать только если используется объединенный список наборов шифров на основе сертификатов **и** PSK. Это происходит в двух случаях: сервер (прокси) с настроенным сертификатом (на сервере наборы шифров PSK всегда настроены, на прокси - если криптографическая библиотека поддерживает PSK), агент настроен на прием входящих соединений как на основе сертификатов, так и на основе PSK
- в остальных случаях достаточно TLSCipherCert* и/или TLSCipherPSK*

В следующих таблицах показаны встроенные значения по умолчанию `TLSCipher*`. Они могут быть хорошей отправной точкой для ваших собственных пользовательских значений.

Параметр	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Параметр	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] Значения по умолчанию отличаются для более старых версий OpenSSL (1.0.1, 1.0.2, 1.1.0), для LibreSSL и если OpenSSL скомпилирован без поддержки PSK.

Примеры наборов шифров, заданных пользователем

Ниже приведены следующие примеры наборов шифров, заданных пользователем:

- Проверка строк шифров и разрешение только наборов шифров PFS
- Переход с AES128 на AES256

Тестирование строк шифров и разрешение только наборов шифров PFS

Чтобы увидеть, какие наборы шифров выбраны, необходимо задать `DebugLevel=4` в файле конфигурации или использовать параметр `-vv` для zabbix_sender.

Перед получением нужных наборов шифров может потребоваться поэкспериментировать с параметрами `TLSCipher*`. Неудобно несколько раз перезапускать сервер Zabbix, прокси или агент только для настройки параметров `TLSCipher*`. Более удобные варианты - использовать zabbix_sender или команду `openssl`. Рассмотрим оба варианта.

1. Использование zabbix_sender.

Создадим тестовый файл конфигурации, например `/home/zabbix/test.conf`, со синтаксисом файла `zabbix_agentd.conf`:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

Для этого примера необходимы действительные сертификаты CA и агента, а также PSK. Измените пути и имена файлов сертификатов и PSK в соответствии с вашей средой.

Если вы не используете сертификаты, а только PSK, можно создать более простой тестовый файл:

```
Hostname=nonexisting
ServerActive=nonexisting
```

```
TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Выбранные наборы шифров можно увидеть, запустив zabbix_sender (пример скомпилирован с OpenSSL 1.1.d):

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

Здесь показаны наборы шифров, выбранные по умолчанию. Эти значения по умолчанию выбраны для обеспечения совместимости с агентами Zabbix, работающими в системах со старыми версиями OpenSSL (начиная с 1.0.1).

В более новых системах можно повысить уровень безопасности, разрешив только несколько наборов шифров, например только наборы шифров с PFS (Perfect Forward Secrecy, совершенная прямая секретность). Настроим параметры TLSCipher*, чтобы разрешить только наборы шифров, поддерживающие PFS.

Attention:
Результат не будет совместим с системами, использующими OpenSSL 1.0.1 и 1.0.2, если используется PSK. Шифрование на основе сертификатов должно работать.

Добавьте две строки в файл конфигурации test.conf:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

и снова выполните тест:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

Списки «certificate ciphersuites» и «PSK ciphersuites» изменились: они стали короче и теперь содержат только наборы шифров TLS 1.3 и наборы шифров TLS 1.2 ECDHE-*, как и ожидалось.

2. TLSCipherAll и TLSCipherAll13 нельзя протестировать с помощью zabbix_sender: они не влияют на значение «certificate and PSK ciphersuites», показанное в примере выше. Для настройки TLSCipherAll и TLSCipherAll13 необходимо экспериментировать с агентом, прокси или сервером.

Таким образом, чтобы разрешить только наборы шифров PFS, может потребоваться добавить до трех параметров

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

в zabbix_agentd.conf, zabbix_proxy.conf и zabbix_server.conf, если для каждого из них настроен сертификат, а для агента также настроен PSK.

Если в вашей среде Zabbix используется только шифрование на основе PSK и сертификаты не используются, достаточно одного параметра:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Теперь, когда вы понимаете принцип работы, можно протестировать выбор наборов шифров даже за пределами Zabbix с помощью команды openssl. Протестируем значения всех трех параметров TLSCipher*:

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

Для получения более подробного вывода можно использовать openssl ciphers с параметром -V:

```
$ openssl ciphers -V EECDH+aRSA+AES128:kECDHEPSK+AES128
           0x13,0x02 - TLS_AES_256_GCM_SHA384  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(256) Mac=AEAD
           0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256
           0x13,0x01 - TLS_AES_128_GCM_SHA256  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(128) Mac=AEAD
```

```
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA256
0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA1
```

Аналогичным образом можно протестировать строки приоритетов для GnuTLS:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

Переход с AES128 на AES256

Zabbix использует AES128 как встроенное значение по умолчанию для данных. Предположим, что вы используете сертификаты и хотите перейти на AES256 в OpenSSL 1.1.1.

Это можно сделать, добавив соответствующие параметры в `zabbix_server.conf`:

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
Хотя будут использоваться только наборы шифров, связанные с сертификатами, параметры `TLSCipherPSK*` также определены, чтобы избежать использования их значений по умолчанию, которые включают менее безопасные наборы шифров ради более широкой совместимости. Наборы шифров PSK нельзя полностью отключить на сервере/прокси.

И в `zabbix_agentd.conf`:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Устранение неполадок с шифрованием Общие рекомендации:

- Начните с понимания того, какой компонент в проблемном случае выступает в роли TLS-клиента, а какой — в роли TLS-сервера.
 Сервер Zabbix, прокси и агенты, в зависимости от взаимодействия между ними, могут работать как TLS-серверы и клиенты.
 Например, сервер Zabbix, подключающийся к агенту для пассивной проверки, выступает как TLS-клиент. Агент в этом случае играет роль TLS-сервера.
 Агент Zabbix, запрашивающий список активных проверок у прокси, выступает как TLS-клиент. Прокси в этом случае играет роль TLS-сервера.
 Утилиты `zabbix_get` и `zabbix_sender` всегда выступают как TLS клиенты.
- Zabbix использует взаимную аутентификацию.
 Каждая сторона проверяет другую и может отклонить соединение.
 Например, сервер Zabbix, подключающийся к агенту, может немедленно закрыть соединение, если сертификат агента недействителен. И наоборот — агент Zabbix, принимающий соединение от сервера, может закрыть соединение, если сервер не доверен агенту.

- Изучите файлы журналов с обеих сторон — у TLS-клиента и у TLS-сервера.
 Сторона, которая отклоняет соединение, может точно указать причину отказа. Другая сторона часто сообщает довольно общее сообщение об ошибке (например, "Connection closed by peer", "connection was non-properly terminated").
- Иногда неверно настроенное шифрование приводит к запутанным сообщениям об ошибках, которые никак не указывают на реальную причину.
 В подразделах ниже мы постараемся привести (далеко не исчерпывающий) набор сообщений и возможных причин, которые могут помочь при устранении неполадок.
 Обратите внимание, что разные криптографические библиотеки (OpenSSL, GnuTLS) часто выдают разные сообщения об ошибках в одних и тех же проблемных ситуациях.
 Иногда сообщения об ошибках зависят даже от конкретной комбинации криптографических библиотек на обеих сторонах.

Устранение неполадок с типом подключения или проблемами с разрешениями Сервер настроен на подключение к агенту с использованием PSK, но агент принимает только незашифрованные соединения

В журнале сервера или прокси (с *GnuTLS* 3.3.16)

```
                    : zbx_tls_connect(): gnutls_handshake() failed: \
-110 TLS-                       .
```

В журнале сервера или прокси (с *OpenSSL* 1.0.2c)

```
                    : TCP-                ,              TLS    [[127.0.0.1]:10050]: \
                      .
```

Одна сторона подключается с сертификатом, а другая принимает только PSK, или наоборот

В любом журнале (*GnuTLS*):

```
                    :   127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
-21                             .
```

В любом журнале (*OpenSSL* 1.0.2c):

```
                    :   127.0.0.1: TLS handshake returned error code 1:\
file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
TLS write fatal alert "handshake failure"
```

Попытка использовать Zabbix sender, скомпилированный с поддержкой TLS, для отправки данных на сервер/прокси Zabbix, скомпилированные без поддержки TLS В журнале на стороне подключающегося:

Linux:

```
...In zbx_tls_init_child()
...       OpenSSL (    OpenSSL 1.1.1  11 Sep 2018)
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...       OpenSSL (    OpenSSL 1.1.1a  20 Nov 2018)
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

В журнале принимающей стороны:

```
...                       : from 127.0.0.1:       TLS
```

Одна сторона подключается с PSK, но другая сторона использует LibreSSL или собрана без поддержки шифрования LibreSSL не поддерживает PSK.

В журнале стороны, выполняющей подключение:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

В журнале принимающей стороны:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

В веб-интерфейсе Zabbix:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Одна сторона подключается с PSK, но другая сторона использует OpenSSL с отключенной поддержкой PSK В журнале стороны, выполняющей подключение:

```
...TCP     ,              TLS   [[192.168.1.2]:10050]: SSL_connect()              SSL_ERROR_SSL: file ../
```

В журнале стороны, принимающей соединение:

```
...                        : from 192.168.1.2: TLS handshake               1: file ssl/statem/statem_srvr.c
```

Шифрование с использованием сертификатов

Обзор Zabbix может использовать RSA-сертификаты в формате PEM, подписанные общедоступным или внутренним центром сертификации (CA).

Проверка сертификата выполняется по предварительно настроенному сертификату CA. При необходимости можно использовать списки отзыва сертификатов (CRL).

Для каждого компонента Zabbix можно настроить только один сертификат.

Дополнительные сведения о настройке и эксплуатации внутреннего CA, создании и подписании запросов на сертификаты, а также об отзыве сертификатов см. в руководствах, например в OpenSSL PKI Tutorial v2.0.

Тщательно продумайте и протестируйте расширения сертификата. Подробнее см. в разделе Ограничения на использование расширений сертификата X.509 v3.

Параметры конфигурации сертификатов Следующие параметры конфигурации поддерживаются для настройки сертификатов на компонентах Zabbix.

Parameter	Mandatory	Description
TLSCAFile	yes	Полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата удаленной стороны. Если используется цепочка сертификатов с несколькими элементами, располагайте сертификаты CA нижнего уровня первыми, а затем сертификаты CA более высокого уровня. Сертификаты от нескольких CA можно включить в один файл.
TLSCRLFile	no	Полный путь к файлу, содержащему списки отзыва сертификатов (CRL).
TLSCertFile	yes	Полный путь к файлу, содержащему сертификат. Если используется цепочка сертификатов с несколькими элементами, располагайте сертификат сервера, прокси или агента первым, затем сертификаты CA нижнего уровня и завершайте сертификатами CA более высокого уровня.
TLSKeyFile	yes	Полный путь к файлу, содержащему закрытый ключ. Убедитесь, что этот файл доступен для чтения только пользователю Zabbix, задав соответствующие права доступа.
TLSServerCertIssuer	no	Допустимый издатель сертификата сервера.
TLSServerCertSubject	no	Допустимый субъект сертификата сервера.

Примеры конфигурации После настройки необходимых сертификатов настройте компоненты Zabbix для использования шифрования на основе сертификатов.

Ниже приведены подробные шаги по настройке:

- сервер Zabbix
- прокси Zabbix
- агент Zabbix

Сервер Zabbix

1. Подготовьте файл сертификатов CA.

Чтобы проверять сертификаты узлов, сервер Zabbix должен иметь доступ к файлу, содержащему самоподписанные корневые сертификаты CA верхнего уровня. Например, если необходимы сертификаты двух независимых корневых CA, поместите их в файл `/home/zabbix/zabbix_ca_file.crt`:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Поместите сертификат или цепочку сертификатов сервера Zabbix в файл, например `/home/zabbix/zabbix_server.crt`. Первым должен идти сертификат сервера Zabbix, за ним - сертификат промежуточного CA:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
```

```
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
Используйте только указанные выше атрибуты для сертификатов клиента и сервера, чтобы не повлиять на процесс проверки сертификатов. Например, OpenSSL может не установить зашифрованное соединение, если используются расширения *X509v3 Subject Alternative Name* или *Netscape Cert Type*. Дополнительную информацию см. в разделе Ограничения при использовании расширений сертификатов X.509 v3.

3. Поместите закрытый ключ сервера Zabbix в файл, например `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Измените параметры конфигурации TLS в файле конфигурации сервера Zabbix:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Прокси Zabbix

1. Подготовьте файлы с корневыми сертификатами CA, сертификатом или цепочкой сертификатов прокси Zabbix и закрытым ключом, как описано в разделе Сервер Zabbix. Затем соответствующим образом измените параметры `TLSCAFile`, `TLSCertFile` и `TLSKeyFile` в файле конфигурации прокси Zabbix.

2. Измените дополнительные параметры TLS в файле конфигурации прокси Zabbix:

- Для активного прокси: `TLSConnect=cert`
- Для пассивного прокси: `TLSAccept=cert`

Note:
Для повышения безопасности прокси можно также задать параметры TLSServerCertIssuer и TLSServerCertSubject. Дополнительную информацию см. в разделе Ограничение допустимых издателя и субъекта сертификата.

Параметры TLS в итоговом файле конфигурации прокси могут выглядеть следующим образом:

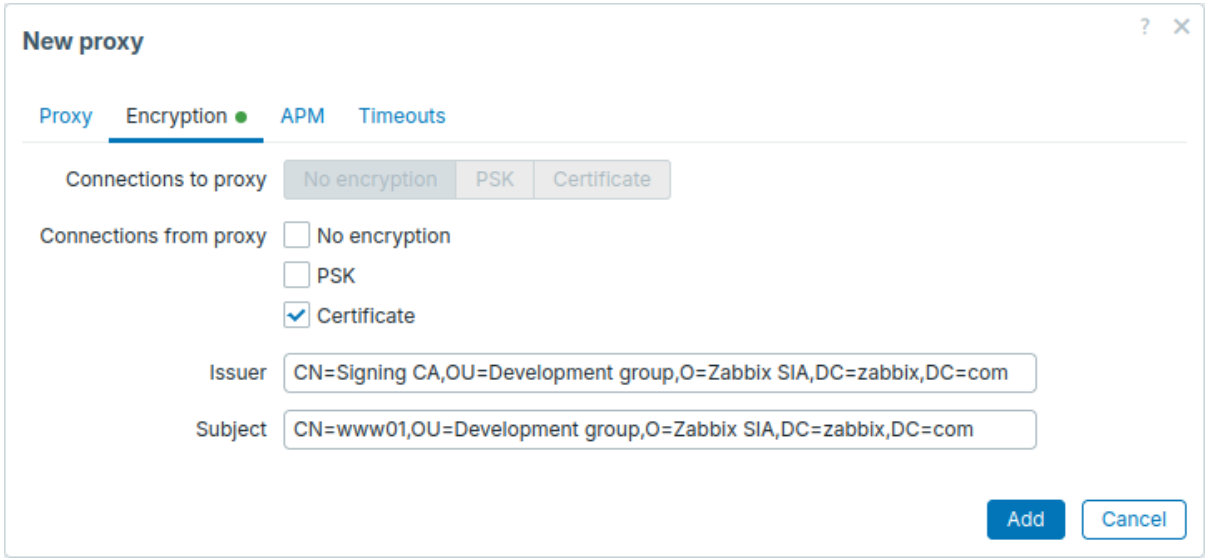
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Настройте шифрование для этого прокси в веб-интерфейсе Zabbix:

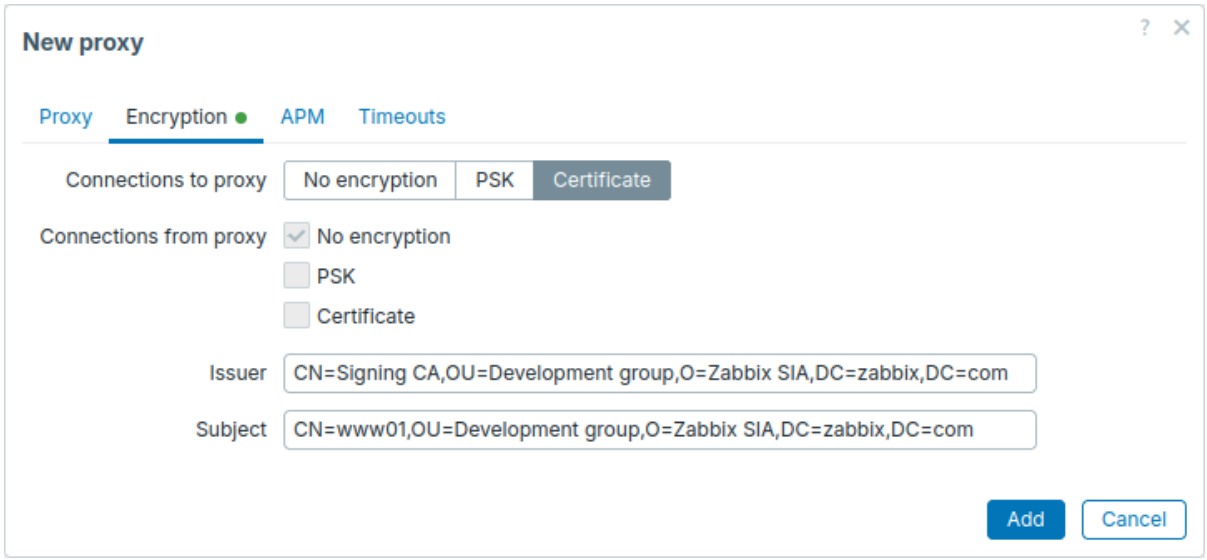
- Перейдите в раздел: *Администрирование* > *Прокси*.
- Выберите прокси и перейдите на вкладку *Шифрование*.

В приведенных ниже примерах поля *Издатель* и *Субъект* заполнены. Дополнительную информацию о назначении этих полей и способах их использования см. в разделе Ограничение допустимых издателя и субъекта сертификата.

Для активного прокси:



Для пассивного прокси:



Агент Zabbix

1. Подготовьте файлы с корневыми сертификатами центра сертификации (CA), сертификатом агента Zabbix или

цепочкой сертификатов, а также закрытым ключом, как описано в разделе Сервер Zabbix. Затем соответствующим образом измените параметры TLSCAFile, TLSCertFile и TLSKeyFile в файле конфигурации агента Zabbix.

2. Измените дополнительные параметры TLS в файле конфигурации агента Zabbix:

- Для активного агента: TLSConnect=cert
- Для пассивного агента: TLSAccept=cert

Note:
Для повышения безопасности агента можно задать параметры TLSServerCertIssuer и TLSServerCertSubject. Дополнительную информацию см. в разделе Ограничение допустимых издателя и субъекта сертификата.

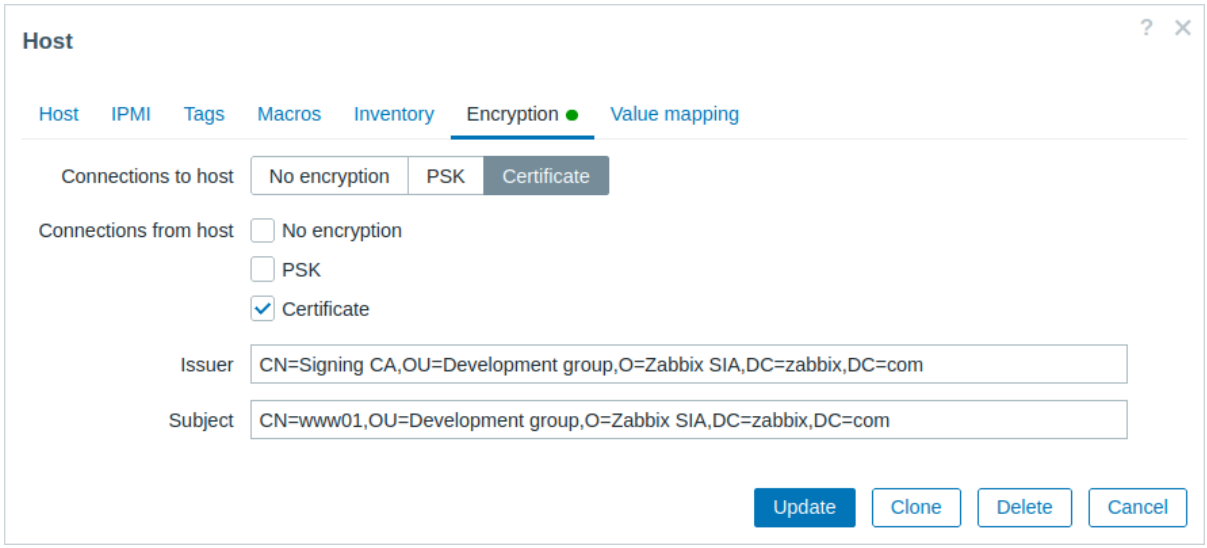
Параметры TLS в итоговом файле конфигурации агента могут выглядеть следующим образом. Обратите внимание, что в примере предполагается, что узел сети отслеживается прокси, поэтому в качестве субъекта сертификата указан прокси:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. Настройте шифрование в веб-интерфейсе Zabbix для узла сети, отслеживаемого этим агентом.

- Перейдите в раздел: *Сбор данных* > *Узлы сети*.
- Выберите узел сети и перейдите на вкладку *Шифрование*.

В примере ниже заполнены поля *Издатель* и *Субъект*. Дополнительную информацию о назначении и использовании этих полей см. в разделе Ограничение допустимых издателя и субъекта сертификата.



Веб-сервис Zabbix

1. Подготовьте файлы с сертификатами корневого центра сертификации, сертификатом/цепочкой сертификатов веб-сервиса Zabbix и закрытым ключом, как описано в разделе Сервер Zabbix. Затем измените параметры TLSCAFile, TLSCertFile и TLSKeyFile в соответствии с этим в файле конфигурации веб-сервиса Zabbix.

2. Измените дополнительный параметр TLS в файле конфигурации веб-сервиса Zabbix: TLSAccept=cert

Параметры TLS в итоговом файле конфигурации веб-сервиса могут выглядеть следующим образом:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Настройте сервер Zabbix для подключения к веб-сервису Zabbix с настроенным TLS, изменив параметр WebServiceURL в файле конфигурации сервера Zabbix:

```
WebServiceURL=https://example.com:443/report
```

Ограничение допустимых издателей и субъектов сертификатов Когда два компонента Zabbix (например, сервер и агент) устанавливают TLS-соединение, они проверяют сертификаты друг друга. Если сертификат узла сети подписан доверенным центром сертификации (с предварительно настроенным корневым сертификатом в TLSCAFile), действителен, не просрочен и проходит другие проверки, обмен данными между компонентами может продолжаться. В этом простейшем случае издатель и субъект сертификата не проверяются.

Однако это создает риск: любой, у кого есть действительный сертификат, может выдавать себя за кого-то другого (например, сертификат узла сети можно использовать для имитации сервера). Это может быть приемлемо в небольших средах, где сертификаты подписываются выделенным внутренним центром сертификации и риск имитации невелик, но может оказаться недостаточным в крупных средах или средах с повышенными требованиями к безопасности.

Если ваш корневой центр сертификации выдает сертификаты, которые не должны приниматься Zabbix, или вы хотите снизить риск имитации, можно ограничить допустимые сертификаты, указав их издателя и субъект.

Например, в файле конфигурации прокси Zabbix можно указать:

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

С этими настройками активный прокси не будет обмениваться данными с сервером Zabbix, сертификат которого имеет другого издателя или субъекта. Аналогично, пассивный прокси не будет принимать запросы от такого сервера.

Правила сопоставления строк Issuer и Subject

Правила сопоставления строк Issuer и Subject следующие:

- Строки Issuer и Subject проверяются независимо. Обе строки необязательны.
- Неуказанная строка означает, что принимается любая строка.
- Строки сравниваются *как есть* и должны совпадать точно.
- Поддерживаются символы UTF-8. Однако подстановочные знаки (*) и регулярные выражения не поддерживаются.
- Реализованы следующие требования RFC 4514 - символы, требующие экранирования (с помощью обратной косой черты '\', U+005C):
  - в любом месте строки: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - в начале строки: пробел (' ', U+0020) или знак номера ('#', U+0023);
  - в конце строки: пробел (' ', U+0020).
- Символы null (U+0000) не поддерживаются. Если встречается символ null, сопоставление завершится неудачей.
- Стандарты RFC 4517 и RFC 4518 не поддерживаются.

Например, если строки организации (O) для Issuer и Subject содержат пробелы в конце, а строка организационного подразделения (OU) для Subject содержит двойные кавычки, эти символы необходимо экранировать:

```
TLSServerCertIssuer=CN=Signing CA,OU=Development head,O=\ Example SIA\ ,DC=example,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group \"5\",O=\ Example SIA\ ,DC=example,DC=com
```

Порядок и форматирование полей

Zabbix следует рекомендациям RFC 4514, в которых для этих полей указан "обратный" порядок: начиная с полей нижнего уровня (CN), переходя к полям среднего уровня (OU, O) и завершая полями высшего уровня (DC).

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

В отличие от этого, OpenSSL по умолчанию отображает строки Issuer и Subject в порядке от верхнего уровня к нижнему. В следующем примере поля Issuer и Subject начинаются с верхнего уровня (DC) и заканчиваются полем нижнего уровня (CN). Форматирование с пробелами и разделителями полей также зависит от используемых параметров, поэтому оно не будет соответствовать формату, требуемому Zabbix.

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Чтобы правильно отформатировать строки *Issuer* и *Subject* для Zabbix, вызовите OpenSSL со следующими параметрами:

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname\
    -in /home/zabbix/zabbix_proxy.crt
```

После этого вывод будет в обратном порядке, с разделением запятыми и пригоден для использования в файлах конфигурации Zabbix и веб-интерфейс:

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

Ограничения на использование расширений сертификатов X.509 v3 При реализации сертификатов X.509 v3 в Zabbix некоторые расширения могут поддерживаться не полностью или приводить к непоследовательному поведению.

Расширение Subject Alternative Name

Zabbix не поддерживает расширение *Subject Alternative Name*, которое используется для указания альтернативных DNS-имен, таких как IP-адреса или адреса электронной почты. Zabbix может проверять только значение в поле *Subject* сертификата (см. Ограничение разрешенных Issuer и Subject сертификата). Если сертификаты содержат поле `subjectAltName`, результат проверки сертификата может различаться в зависимости от конкретных криптографических библиотек, использованных при сборке компонентов Zabbix. В результате Zabbix может как принять, так и отклонить сертификаты в зависимости от этой комбинации.

Расширение Extended Key Usage

Zabbix поддерживает расширение *Extended Key Usage*. Однако при его использовании, как правило, требуется указывать оба атрибута: *clientAuth* (для аутентификации TLS WWW client) и *serverAuth* (для аутентификации TLS WWW server). Например:

- При пассивных проверках, где Zabbix агент работает как TLS-сервер, атрибут *serverAuth* должен быть включен в сертификат агента.
- Для активных проверок, где агент работает как TLS-клиент, атрибут *clientAuth* должен быть включен в сертификат агента.

Хотя GnuTLS может выдавать предупреждение о нарушениях использования ключа, обычно он все равно позволяет продолжить обмен данными, несмотря на эти предупреждения.

Расширение Name Constraints

Поддержка расширения *Name Constraints* различается в зависимости от криптографических библиотек. Убедитесь, что выбранная вами библиотека поддерживает это расширение. Это расширение может не позволить Zabbix загружать CA-сертификаты, если этот раздел помечен как критический, в зависимости от используемой библиотеки.

Списки отзыва сертификатов (CRL) Если сертификат скомпрометирован, Центр сертификации (CA) может отозвать его, включив сертификат в Список отзыва сертификатов (CRL). CRL управляются через файлы конфигурации и могут быть указаны с помощью параметра `TLSCRLFile` в файлах конфигурации сервера, прокси и агента. Например:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

В этом случае `zabbix_crl_file.crt` может содержать CRL от нескольких CA и может выглядеть так:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

Файл CRL загружается только при запуске Zabbix. Чтобы обновить CRL, перезапустите Zabbix.

> **Attention:**
> Если компоненты Zabbix скомпилированы с OpenSSL и используются CRL, убедитесь, что для каждого корневого и промежуточного CA в цепочках сертификатов есть соответствующий CRL (даже если он пустой), включенный в `TLSCRLFile`.

Устранение неполадок с сертификатами OpenSSL используется с CRL, и для некоторого CA в цепочке сертификатов его CRL не включен в TLSCRLFile

В журнале сервера TLS в случае узла-пира *OpenSSL*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

В журнале сервера TLS в случае узла-пира *GnuTLS*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL истек или истекает во время работы сервера

OpenSSL, в журнале сервера:

- до истечения срока действия:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- после истечения срока действия:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

Суть здесь в том, что при действительном CRL отозванный сертификат сообщается как ”certificate revoked”. Когда CRL истекает, сообщение об ошибке меняется на ”certificate expired”, что довольно вводит в заблуждение.

GnuTLS, в журнале сервера:

- до и после истечения срока действия одно и то же:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

Самоподписанный сертификат, неизвестный CA

OpenSSL, в журнале:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

Это наблюдалось, когда сертификат сервера по ошибке имел одинаковые строки Issuer и Subject, хотя был подписан CA. Issuer и Subject равны в корневом сертификате CA верхнего уровня, но они не могут быть равны в сертификате сервера. (То же относится к сертификатам прокси и агента.)

Чтобы проверить, содержит ли сертификат одинаковые записи Issuer и Subject, выполните:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Для корневого сертификата верхнего уровня допустимо, чтобы значения Issuer и Subject были одинаковыми.

Шифрование с использованием предварительно общих ключей

Обзор Каждый предварительно установленный ключ (PSK) в Zabbix фактически представляет собой пару:

- несекретная строка идентификатора PSK;
- секретное строковое значение PSK.

Строка идентификатора PSK представляет собой непустую строку в кодировке UTF-8. Например: ”PSK ID 001 Zabbix agentd”. Это уникальное имя, по которому компоненты Zabbix обращаются к этому конкретному PSK. Не указывайте конфиденциальную информацию в строке идентификатора PSK — она передается по сети в незашифрованном виде.

Значение PSK — это строка из шестнадцатеричных цифр, которую сложно угадать, например: `e560cb0d918d26d31b4f642181f5f57`

Warning:
Никогда не используйте приведенные в документации примеры значений PSK в производственных средах. Вместо этого сгенерируйте уникальный криптографически стойкий PSK.

Ограничения размера В Zabbix существуют ограничения размера идентификатора PSK и значения PSK; в некоторых случаях криптографическая библиотека может иметь более низкое ограничение:

Компонент	Максимальный размер идентификатора PSK	Минимальный размер значения PSK	Максимальный размер значения PSK
Zabbix	128 символов UTF-8	128 бит (PSK размером 16 байт, вводится как 32 шестнадцатеричных цифры)	2048 бит (PSK размером 256 байт, вводится как 512 шестнадцатеричных цифр)
GnuTLS	128 байт (может включать символы UTF-8)	-	2048 бит (PSK размером 256 байт, вводится как 512 шестнадцатеричных цифр)
OpenSSL 1.0.x, 1.1.0	127 байт (может включать символы UTF-8)	-	2048 бит (PSK размером 256 байт, вводится как 512 шестнадцатеричных цифр)
OpenSSL 1.1.1	127 байт (может включать символы UTF-8)	-	512 бит (PSK размером 64 байта, вводится как 128 шестнадцатеричных цифр)
OpenSSL 1.1.1a и более поздние версии	127 байт (может включать символы UTF-8)	-	2048 бит (PSK размером 256 байт, вводится как 512 шестнадцатеричных цифр)

Attention:
Веб-интерфейс Zabbix позволяет настроить строку идентификатора PSK длиной до 128 символов и PSK длиной до 2048 бит независимо от используемых криптографических библиотек.
 Если некоторые компоненты Zabbix поддерживают более низкие ограничения, пользователь должен самостоятельно настроить идентификатор PSK и значение PSK с допустимой для этих компонентов длиной.
 Превышение ограничений размера приводит к сбоям связи между компонентами Zabbix.

Перед подключением сервера Zabbix к агенту с использованием PSK сервер ищет в базе данных (фактически в кэше конфигурации) идентификатор PSK и значение PSK, настроенные для этого агента. При получении подключения агент использует идентификатор PSK и значение PSK из своего файла конфигурации. Если у обеих сторон совпадают строка идентификатора PSK и значение PSK, подключение может быть установлено.

Attention:
Каждый идентификатор PSK должен быть связан только с одним значением. Пользователь должен убедиться, что не существует двух PSK с одинаковой строкой идентификатора, но разными значениями. Несоблюдение этого требования может привести к непредсказуемым ошибкам или нарушениям связи между компонентами Zabbix, использующими PSK с этой строкой идентификатора PSK.

Генерация PSK Например, PSK длиной 256 бит (32 байта) можно сгенерировать с помощью следующих команд:

- с помощью *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- с помощью *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Обратите внимание, что команда "psktool" выше создает файл базы данных с идентификатором PSK и соответствующим ему PSK. Zabbix ожидает, что файл PSK будет содержать только PSK, поэтому идентификатор и двоеточие (':') необходимо удалить из файла.

Настройка PSK для взаимодействия сервера и агента (пример) На узле сети с агентом запишите сгенерированное выше значение PSK в файл, например `/home/zabbix/zabbix_agentd.psk`. В первой строке файла должно содержаться значение PSK.

Настройте права доступа к файлу PSK: он должен быть доступен для чтения только пользователю Zabbix.

Измените параметры TLS в файле конфигурации агента Zabbix (`zabbix_agentd.conf`). Например, задайте:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

Агент будет подключаться к серверу (активные проверки) и принимать от сервера и `zabbix_get` только подключения с использованием PSK. Идентификатором PSK будет значение "PSK 001".

Перезапустите агент. Теперь можно проверить подключение с помощью `zabbix_get`, например:

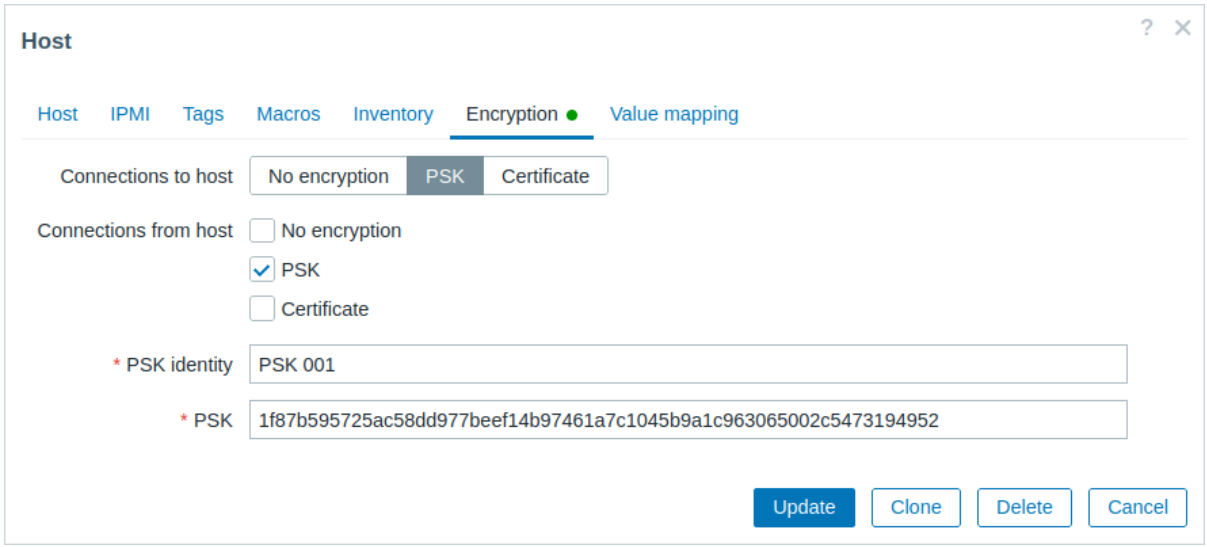
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Чтобы свести время простоя к минимуму, см. информацию об изменении типа подключения в разделе Управление шифрованием подключений).

Настройте шифрование PSK для этого агента в веб-интерфейсе Zabbix:

1. Перейдите в раздел: *Сбор данных* > *Узлы сети*.
2. Выберите узел сети и перейдите на вкладку *Шифрование*.
3. Настройте параметры шифрования PSK.

Пример:



Все обязательные поля ввода отмечены красной звездочкой.

После синхронизации кэша конфигурации с базой данных новые подключения будут использовать PSK. Проверьте файлы журналов сервера и агента на наличие сообщений об ошибках.

Настройка PSK для обмена данными между сервером и активным прокси (пример) На прокси запишите значение PSK в файл, например `/home/zabbix/zabbix_proxy.psk`. В первой строке файла должно содержаться значение PSK, например:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Настройте права доступа к файлу PSK: он должен быть доступен для чтения только пользователю Zabbix.

Измените параметры TLS в файле конфигурации прокси Zabbix (`zabbix_proxy.conf`). Например, задайте:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Прокси подключится к серверу с использованием PSK. Идентификатором PSK будет «PSK 002».

(Чтобы минимизировать время простоя, см. описание изменения типа подключения в разделе Управление шифрованием подключений.)

Настройте PSK для этого прокси в веб-интерфейсе Zabbix. Перейдите в раздел *Администрирование* > *Прокси*, выберите прокси и откройте вкладку *Шифрование*. В разделе «Подключения от прокси» установите флажок PSK. Вставьте в поле «Идентификатор PSK» значение «PSK 002», а в поле «PSK» — значение «e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08 Нажмите «Обновить».

Перезапустите прокси. Он начнет использовать зашифрованные подключения к серверу на основе PSK. Проверьте файлы журналов сервера и прокси на наличие сообщений об ошибках.

Для пассивного прокси процедура практически такая же. Единственное отличие заключается в том, что в файле конфигурации прокси Zabbix необходимо задать TLSAccept=psk, а в веб-интерфейсе Zabbix для параметра *Подключения к прокси* выбрать PSK.

Устранение неполадок с PSK PSK содержит нечетное количество шестнадцатеричных цифр

Прокси или агент не запускается, сообщение в журнале прокси или агента:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

Строка идентификатора PSK длиной более 128 байт передается в GnuTLS

В журнале клиента TLS:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

В журнале сервера TLS:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Слишком длинное значение PSK, используемое с OpenSSL 1.1.1

В журнале на стороне подключения:

```
...       OpenSSL (    OpenSSL 1.1.1  11 Sep 2018)
...
...  zbx_tls_connect(): psk_identity:"PSK 1"
...zbx_psk_client_cb()              PSK "PSK 1"
...       zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\ext
```

В журнале на стороне принимающего:

```
...         123.123.123.123             .               .
```

Эта проблема обычно возникает при обновлении OpenSSL с 1.0.x или 1.1.0 до 1.1.1, если значение PSK длиннее 512 бит (PSK длиной 64 байта, введенный как 128 шестнадцатеричных цифр).

См. также: Ограничения размера значения

Безопасное подключение к базе данных

Обзор

В этом разделе приведены шаги настройки Zabbix и примеры конфигурации для защищенных TLS-соединений между:

Database	Zabbix components
MySQL	веб-интерфейс Zabbix, сервер Zabbix, прокси Zabbix
PostgreSQL	веб-интерфейс Zabbix, сервер Zabbix, прокси Zabbix

Чтобы настроить шифрование соединения в СУБД, см. официальную документацию поставщика для получения подробной информации:

- MySQL: серверы баз данных репликации source и replica.
- MySQL: серверы баз данных group replication и т. д.
- PostgreSQL параметры шифрования.

Все примеры основаны на GA-релизах MySQL CE (8.0) и PgSQL (13), доступных через официальные репозитории, с использованием CentOS 8.

Требования

Для настройки шифрования требуется следующее:

- Поддерживаемая разработчиком операционная система с OpenSSL >=1.1.X или альтернативой.

Note:
Рекомендуется избегать использования ОС в статусе end-of-life, особенно в случае новых установок

- Движок базы данных (RDBMS), установленный и поддерживаемый из официального репозитория, предоставляемого разработчиком. Операционные системы часто поставляются с устаревшими версиями программного обеспечения базы данных, для которых поддержка шифрования не реализована, например системы на базе RHEL 7 и PostgreSQL 9.2, MariaDB 5.5 без поддержки шифрования.

Терминология

Установка этой опции принудительно использует TLS-соединение с базой данных со стороны сервера/прокси Zabbix и веб-интерфейса при подключении к базе данных:

- `required` - подключение с использованием TLS в качестве режима транспортировки без проверки идентичности
- `verify_ca` - подключение с использованием TLS и проверка сертификата
- `verify_full` - подключение с использованием TLS, проверка сертификата и проверка того, что идентичность базы данных (CN), указанная в DBHost, соответствует ее сертификату

Конфигурация Zabbix

Веб-интерфейс и база данных

Безопасное подключение к базе данных можно настроить во время установки веб-интерфейса:

- Установите флажок *Database TLS encryption* на шаге Configure DB connection, чтобы включить шифрование передачи данных.
- Установите флажок *Verify database certificate*, который появляется после выбора поля *TLS encryption*, чтобы включить шифрование с использованием сертификатов.

Note:
Для MySQL флажок *Database TLS encryption* отключен, если в поле *Database host* указано значение localhost, поскольку подключение с использованием файла сокета (в Unix) или общей памяти (в Windows) невозможно зашифровать.
 Для PostgreSQL флажок *TLS encryption* отключен, если значение поля *Database host* начинается с косой черты или поле пусто.

В режиме шифрования TLS с сертификатами становятся доступны следующие параметры (если установлены оба флажка):

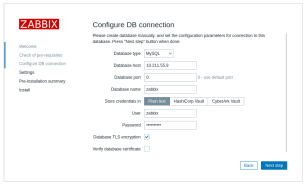
Parameter	Description
Database TLS CA file	Укажите полный путь к действительному файлу центра сертификации (CA) TLS.
Database TLS key file	Укажите полный путь к действительному файлу ключа TLS.
Database TLS certificate file	Укажите полный путь к действительному файлу сертификата TLS.
Database host verification	Установите этот флажок, чтобы активировать проверку узла сети. Отключено для MySQL, поскольку библиотека PHP MySQL не позволяет пропустить этап проверки сертификата узла-партнера.
Database TLS cipher list	Укажите пользовательский список допустимых шифров. Формат списка шифров должен соответствовать стандарту OpenSSL. Доступно только для MySQL.

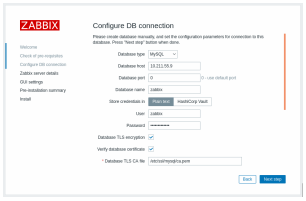
Attention:
Параметры TLS должны указывать на действительные файлы. Если они указывают на несуществующие или недействительные файлы, это приведет к ошибке авторизации.
 Если файлы сертификатов доступны для записи, веб-интерфейс выводит предупреждение в отчете System information: "TLS certificate files must be read-only." (отображается только в том случае, если пользователем сертификата является пользователь PHP).
Сертификаты, защищенные паролями, не поддерживаются.

Варианты использования

Веб-интерфейс Zabbix использует графический интерфейс для определения возможных параметров: `required`, `verify_ca`, `verify_full`. Укажите необходимые параметры на шаге мастера установки *Настройка подключений к БД*. Эти параметры сопоставляются с файлом конфигурации веб-интерфейса (`zabbix.conf.php`) следующим образом:

Настройки графического интерфейса	Файл конфигурации	Описание	Результат

	...
	// Used for TLS connection.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Установите флажок *Шифрование TLS для базы данных*
		2. Не устанавливайте флажок *Проверять сертификат базы данных*
			Включение режима `required`.

	...
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
		2. Укажите путь к *файлу центра сертификации TLS базы данных*
			Включение режима `verify_ca`.

Настройки графического интерфейса	Файл конфигурации	Описание	Результат

	...
	// Used for TLS connection with strictly defined Cipher list.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '<cipher_list>';`
	...
	
	Или:
	
	...
	// Used for TLS connection without Cipher list defined - selected by MySQL server
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
		2. Укажите путь к *файлу ключа TLS базы данных*
		3. Укажите путь к *файлу центра сертификации TLS базы данных*
		4. Укажите путь к *файлу сертификата TLS базы данных*
		5. Укажите *список шифров TLS базы данных* (необязательно)
			Включение режима `verify_full` для MySQL.

Настройки графического интерфейса	Файл конфигурации	Описание	Результат

	...
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = ' ';`
	...
		1. Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
		2. Укажите путь к *файлу ключа TLS базы данных*
		3. Укажите путь к *файлу центра сертификации TLS базы данных*
		4. Укажите путь к *файлу сертификата TLS базы данных*
		5. Установите флажок *Проверка узла сети базы данных*
			Включение режима `verify_full` для PostgreSQL.

См. также: Примеры настройки шифрования для MySQL, Примеры настройки шифрования для PostgreSQL.

Конфигурация сервера/прокси Zabbix

Защищенные подключения к базе данных можно настроить с помощью соответствующих параметров в конфигурационном файле сервера Zabbix и/или прокси.

Конфигурация	Результат
Не задано	Подключение к базе данных без шифрования.
Задать `DBTLSConnect=required`	Сервер/прокси устанавливает TLS-подключение к базе данных. Незашифрованное подключение не разрешено.
1. Задать `DBTLSConnect=verify_ca` 2. Задать `DBTLSCAFile` - указать файл центра сертификации TLS	Сервер/прокси устанавливает TLS-подключение к базе данных после проверки сертификата базы данных.
1. Задать `DBTLSConnect=verify_full` 2. Задать `DBTLSCAFile` - указать файл центра сертификации TLS	Сервер/прокси устанавливает TLS-подключение к базе данных после проверки сертификата базы данных и идентичности узла сети базы данных.
1. Задать `DBTLSCAFile` - указать файл центра сертификации TLS 2. Задать `DBTLSCertFile` - указать файл сертификата открытого ключа клиента 3. Задать `DBTLSKeyFile` - указать файл закрытого ключа клиента	Сервер/прокси предоставляет сертификат клиента при подключении к базе данных.
Задать `DBTLSCipher` - список шифров шифрования, разрешенных клиентом для подключений с использованием протоколов TLS до TLS 1.2 включительно или `DBTLSCipher13` - список шифров шифрования, разрешенных клиентом для подключений с использованием протокола TLS 1.3	(MySQL) TLS-подключение устанавливается с использованием шифра из указанного списка. (PostgreSQL) Задание этого параметра будет считаться ошибкой.

Конфигурация шифрования MySQL

Обзор

В этом разделе приведены несколько примеров настройки шифрования для CentOS 8.2 и MySQL 8.4.0, которые можно использовать как краткое руководство по шифрованию соединения с базой данных.

Attention:
Если для узла MySQL задано значение localhost, параметры шифрования будут недоступны. В этом случае соединение между веб-интерфейсом Zabbix и базой данных использует файл сокета (в Unix) или разделяемую память (в Windows) и не может быть зашифровано.

Note:
Список комбинаций шифрования не ограничивается перечисленными на этой странице. Доступно гораздо больше комбинаций.

Предварительные требования

Установите базу данных MySQL из официального репозитория.

См. документацию MySQL для получения подробной информации о том, как использовать репозиторий MySQL.

Сервер MySQL готов принимать защищенные подключения с использованием самоподписанного сертификата.

Чтобы увидеть, какие пользователи используют зашифрованное соединение, выполните следующий запрос (Performance Schema должен быть включен):

```
SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, processli
FROM performance_schema.status_by_thread  AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

Шифрование только для передачи данных

Конфигурация MySQL

Современные версии базы данных готовы к работе «из коробки» в режиме шифрования, который `required`. Сертификат на стороне сервера будет создан после первоначальной настройки и запуска.

Создайте пользователей и роли для основных компонентов (замените <strong_password> на надежный уникальный пароль):

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

Обратите внимание, что протокол X.509 не используется для проверки личности, однако для пользователя настроено использование только зашифрованных соединений. Дополнительную информацию о настройке пользователей см. в документации MySQL.

Выполните следующую команду для проверки соединения (сокетное соединение нельзя использовать для проверки защищенных соединений):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Проверьте текущий статус и доступные наборы шифров:

```
mysql> status
--------------
mysql Ver 8.4.0 for Linux on x86_64 (MySQL Community Server - GPL)
```

```
Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384

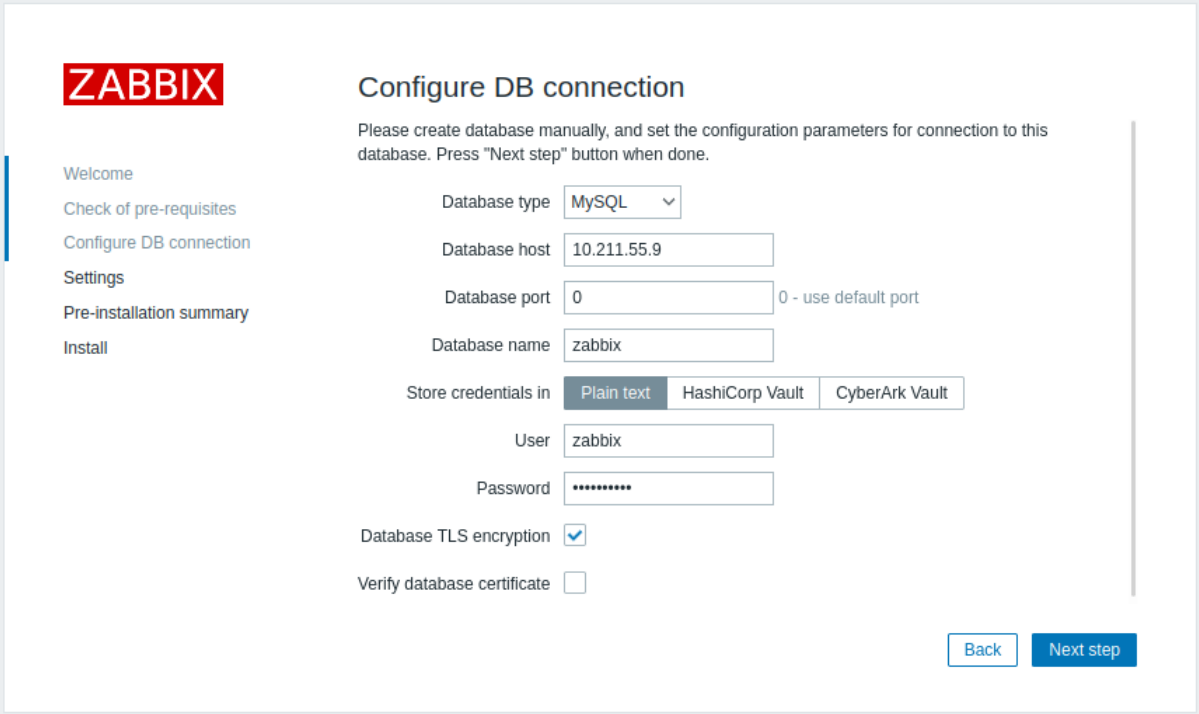

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Веб-интерфейс

Чтобы включить шифрование только для передачи данных для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите *Database TLS encryption*
- Оставьте *Verify database certificate* неотмеченным



Сервер

Чтобы включить шифрование только транспортного уровня для соединений между сервером и базой данных, отредактируйте файл конфигурации сервера:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Шифрование с проверкой центра сертификации

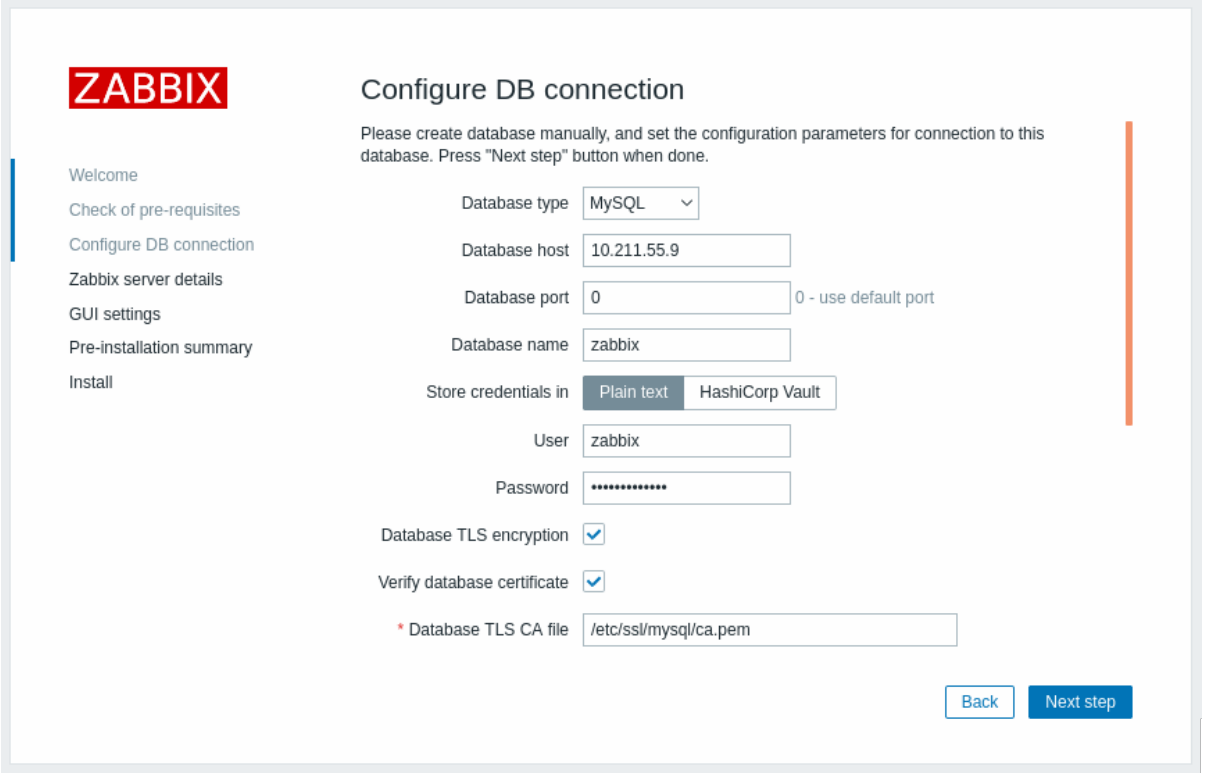
Скопируйте необходимый CA MySQL на сервер веб-интерфейса Zabbix и назначьте соответствующие права доступа, чтобы веб-сервер мог читать этот файл.

Note:
Этот режим не работает в RHEL 7 из-за более старых библиотек MySQL.

Веб-интерфейс

Чтобы включить шифрование с проверкой сертификата для подключений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
- Укажите путь к файлу центра сертификации TLS базы данных



Также это можно настроить в файле конфигурации веб-интерфейса (`zabbix.conf.php`):

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Для устранения неполадок используйте инструмент командной строки, чтобы проверить возможность подключения от имени необходимого пользователя:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Сервер

Чтобы включить шифрование с проверкой сертификата для соединений между сервером Zabbix и базой данных, измените файл конфигурации сервера:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Шифрование с полной проверкой

Конфигурация MySQL

Установите параметр конфигурации сервера MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) в значение:

```
[mysqld]
...
####                            datadir MySQL CE
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
...
```

Ключи для сервера MySQL CE и клиента (веб-интерфейс Zabbix) должны быть созданы вручную в соответствии с документацией MySQL CE: Creating SSL and RSA certificates and keys using MySQL или Creating SSL certificates and keys using openssl

Attention:
Сертификат сервера MySQL должен содержать поле Common Name, установленное в значение FQDN, так как веб-интерфейс Zabbix будет использовать DNS-имя для связи с базой данных или IP-адрес узла сети базы данных.

Создайте пользователя MySQL:

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

Проверьте, можно ли войти с этим пользователем:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

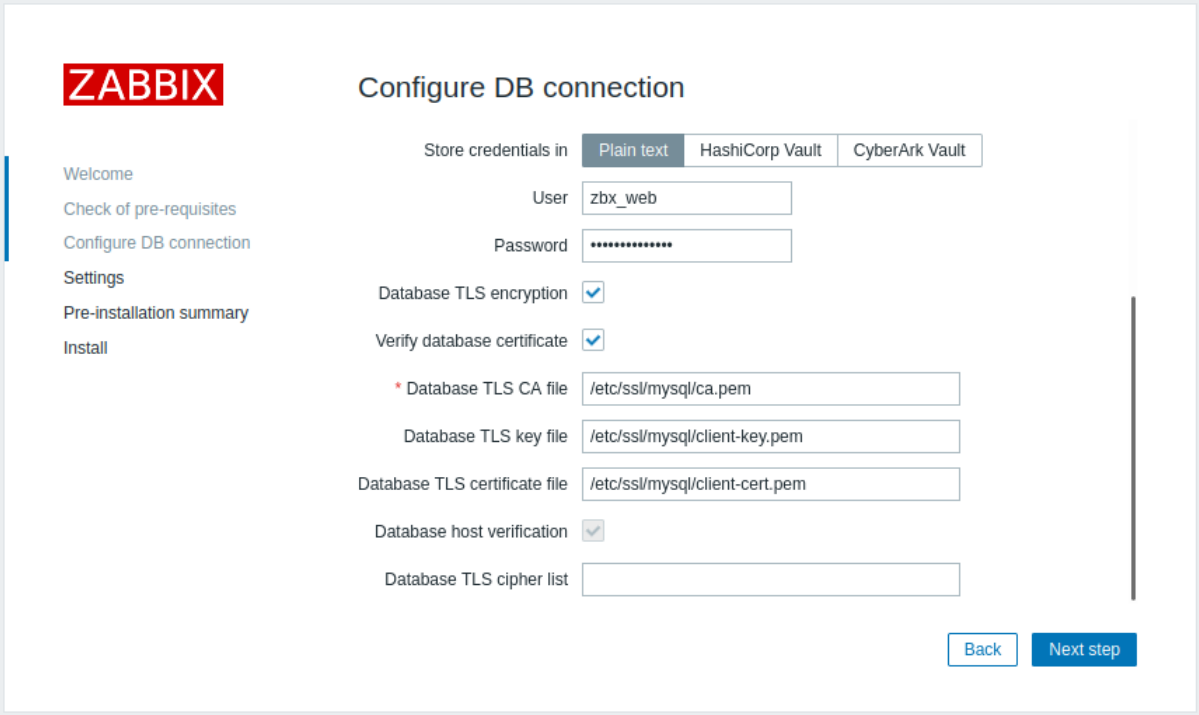
Веб-интерфейс

Чтобы включить шифрование с полной проверкой соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
- Укажите путь к *файлу ключа TLS для базы данных*
- Укажите путь к *файлу центра сертификации TLS для базы данных*
- Укажите путь к *файлу сертификата TLS для базы данных*

Обратите внимание, что флажок *Проверка узла сети базы данных* установлен и недоступен для изменения — для MySQL этот шаг нельзя пропустить.

Warning:
Если поле *Список шифров TLS для базы данных* оставить пустым, будут включены общие шифры, разрешенные и веб-интерфейсом (клиентом), и сервером. В качестве альтернативы шифры можно задать явно в соответствии с требованиями к настройке шифров.



В качестве альтернативы это можно настроить в файле конфигурации веб-интерфейса (`zabbix.conf.php`):

```
...
// Used for TLS connection with strictly defined Cipher list.
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// or
...
// Used for TLS connection without Cipher list defined - selected by MySQL server
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Сервер

Чтобы включить шифрование с полной проверкой для соединений между сервером Zabbix и базой данных, отредактируйте файл конфигурации сервера:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

Настройка шифрования PostgreSQL

Обзор

В этом разделе приведено несколько примеров настройки шифрования для CentOS 8.2 и PostgreSQL 13.

Note:
Соединение между веб-интерфейсом Zabbix и PostgreSQL не может быть зашифровано (параметры в GUI отключены), если значение поля *Database host* начинается со слэша или поле пустое.

Предварительные требования

Установите базу данных PostgreSQL, используя официальный репозиторий.

PostgreSQL по умолчанию не настроен на прием TLS-соединений. Следуйте инструкциям из документации PostgreSQL по подготовке сертификатов с помощью postgresql.conf, а также по контролю доступа пользователей через `ph_hba.conf`.

По умолчанию сокет PostgreSQL привязан к localhost, поэтому для удаленных сетевых подключений необходимо разрешить прослушивание на реальном сетевом интерфейсе.

Параметры PostgreSQL для всех режимов могут выглядеть так:

`/var/lib/pgsql/13/data/postgresql.conf`:

```
...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
ssl_min_protocol_version = 'TLSv1.3'
...
```

Для контроля доступа измените `/var/lib/pgsql/13/data/pg_hba.conf`:

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

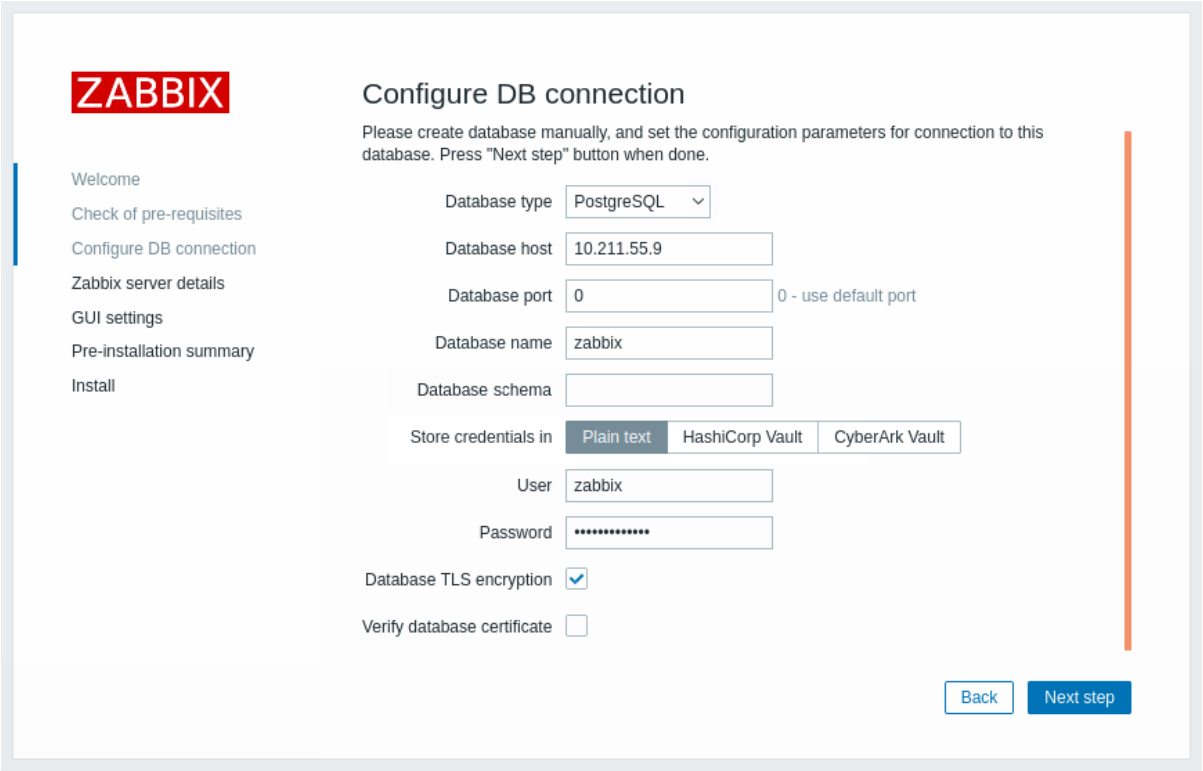
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Шифрование только для передачи данных

Веб-интерфейс

Чтобы включить шифрование только для передачи данных для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажок *Database TLS encryption*
- Оставьте *Verify database certificate* снятым



Сервер

Чтобы включить шифрование только транспортного уровня для соединений между сервером и базой данных, отредактируйте файл конфигурации сервера:

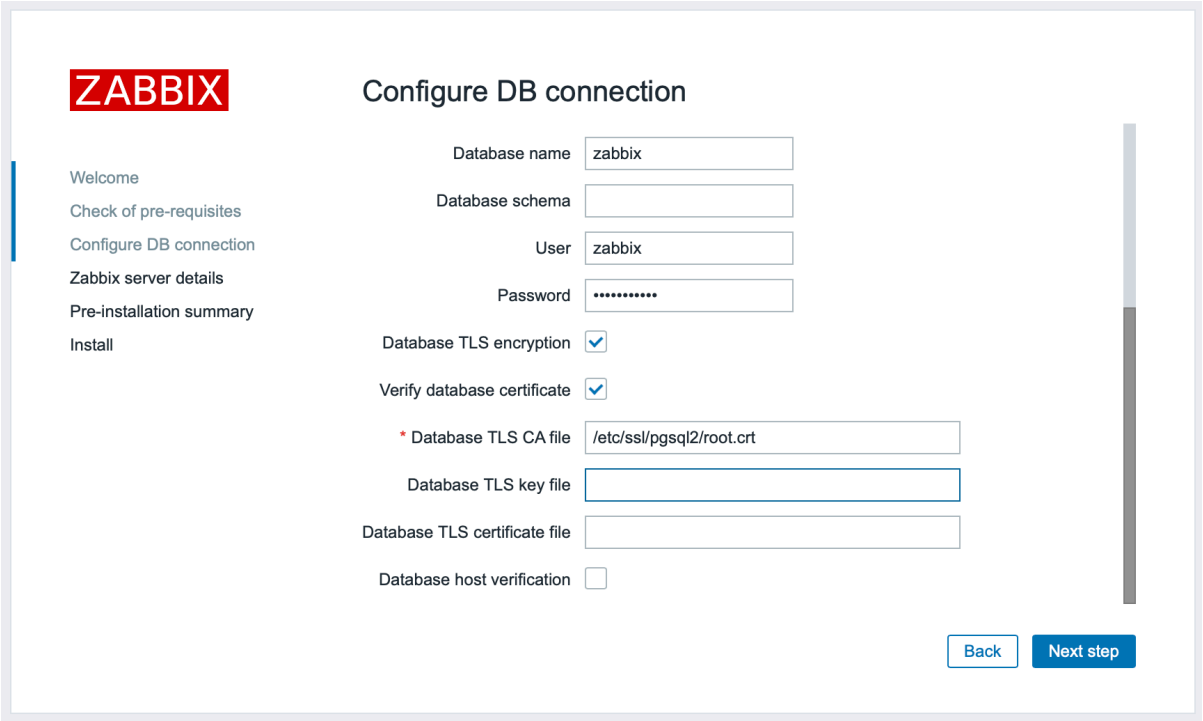
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Шифрование с проверкой центра сертификации

Веб-интерфейс

Чтобы включить шифрование с проверкой центра сертификации для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Шифрование TLS для базы данных* и *Проверять сертификат базы данных*
- Укажите путь к *файлу центра сертификации TLS для базы данных*



Также это можно настроить в файле конфигурации веб-интерфейса (`zabbix.conf.php`):

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Сервер

Чтобы включить шифрование с проверкой сертификата для соединений между сервером Zabbix и базой данных, отредактируйте файл конфигурации сервера (замените `<strong_password>` на надежный уникальный пароль):

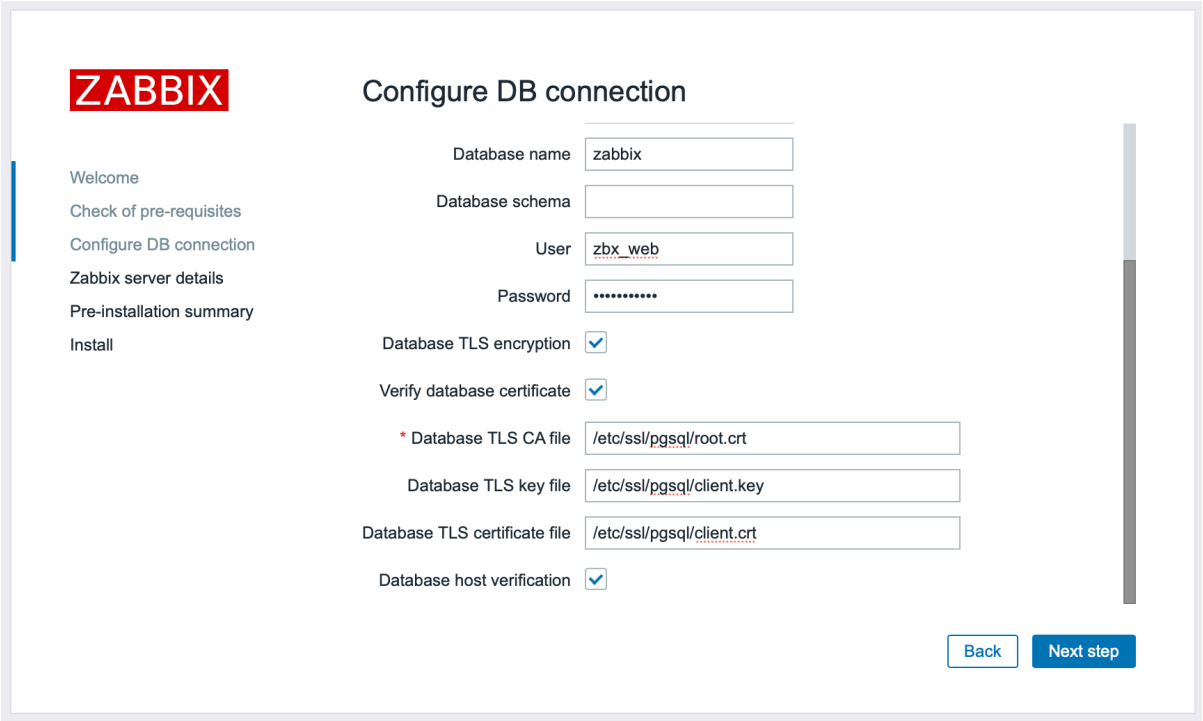
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Шифрование с полной проверкой

Веб-интерфейс

Чтобы включить шифрование с помощью сертификата и проверку идентичности узла базы данных для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Шифрование TLS базы данных* и *Проверять сертификат базы данных*
- Укажите путь к *файлу ключа TLS базы данных*
- Укажите путь к *файлу центра сертификации TLS базы данных*
- Укажите путь к *файлу сертификата TLS базы данных*
- Установите флажок *Проверять узел базы данных*



Также это можно настроить в файле конфигурации веб-интерфейса (`zabbix.conf.php`):

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Сервер

Чтобы включить шифрование с проверкой сертификата и идентификации узла базы данных для подключений между сервером Zabbix и базой данных, отредактируйте файл конфигурации сервера:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

Защищенное соединение между веб-интерфейсом и сервером

Обзор

В этом разделе приведены инструкции по настройке Zabbix и примеры конфигурации для защищенных TLS-соединений между веб-интерфейсом Zabbix и сервером Zabbix.

Настройка

По умолчанию обмен данными между веб-интерфейсом Zabbix и сервером Zabbix не шифруется. Для повышения безопасности включите TLS с обеих сторон. Ниже приведен пример самого простого способа сделать это.

1. Создайте сертификаты и ключи.

Создайте рабочий каталог:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Создайте сертификат CA (измените значение MyZabbixCA в соответствии с фактическим общим именем):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Создайте закрытый ключ и сертификат для сервера Zabbix (измените значение zabbix-server.example.com в соответствии с фактическим общим именем):

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Создайте закрытый ключ и сертификат для веб-интерфейса Zabbix (измените значение zabbix-frontend-node в соответствии с фактическим общим именем):

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. Установите соответствующие разрешения.

Для сервера Zabbix (измените владельца и группу в соответствии с пользователем демона сервера Zabbix в вашем дистрибутиве):

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

Для веб-интерфейса (измените владельца и группу в соответствии с пользователем веб-сервера в вашем дистрибутиве):

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Настройте сервер Zabbix.

В файле конфигурации сервера (zabbix_server.conf) добавьте:

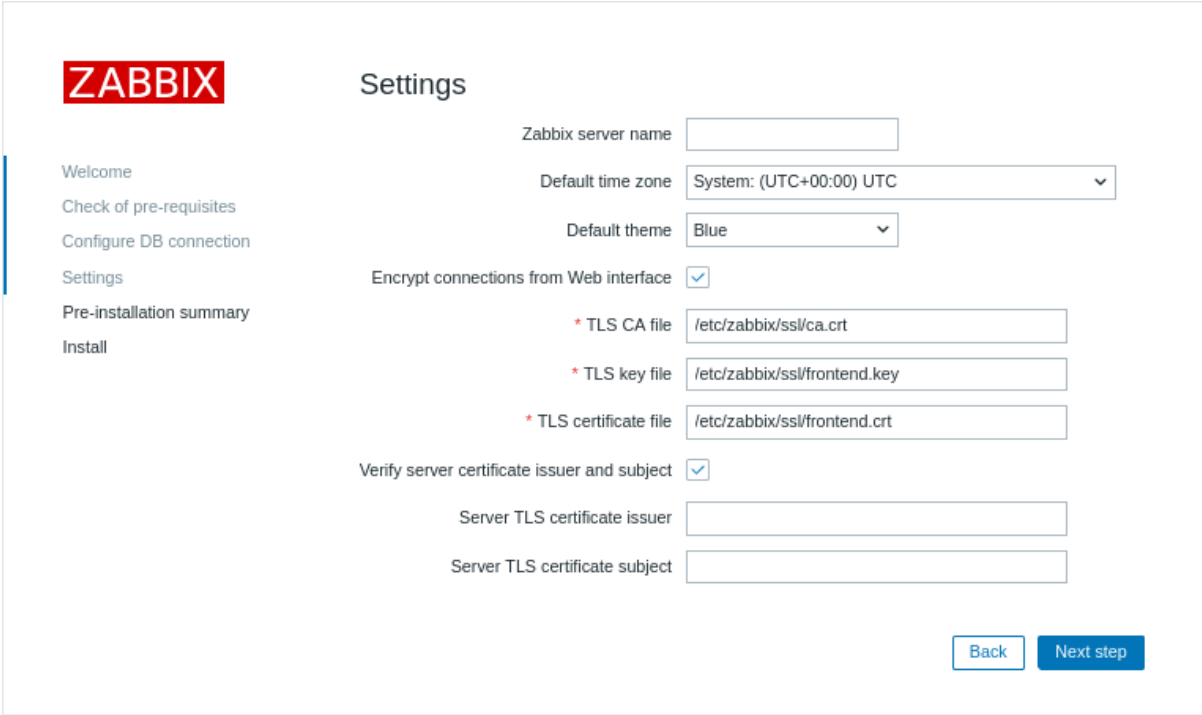
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
###         :
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

Затем перезапустите сервер:

```
sudo systemctl restart zabbix-server
```

4. Настройте веб-интерфейс Zabbix.

Во время установки веб-интерфейса включите параметр *Шифровать соединения из веб-интерфейса* (при необходимости также параметр *Проверять издателя и субъект сертификата сервера*) и заполните поля *Файл TLS CA*, *Файл ключа TLS*, *Файл сертификата TLS* (при необходимости также поля *Издатель сертификата TLS сервера* и *Субъект сертификата TLS сервера*):



Параметр	Описание
Файл TLS CA	Укажите полный путь к файлу сертификата центра сертификации (CA), используемого для проверки сертификата сервера.
Файл ключа TLS	Укажите полный путь к файлу закрытого ключа клиента, соответствующего сертификату клиента.
Файл сертификата TLS	Укажите полный путь к файлу сертификата клиента, если требуется взаимная аутентификация TLS.
Издатель сертификата TLS сервера	Укажите отличительное имя (DN) издателя для сопоставления с сертификатом сервера.
Субъект сертификата TLS сервера	Укажите отличительное имя (DN) субъекта для сопоставления с сертификатом сервера.

В существующих установках измените следующие параметры в файле конфигурации веб-интерфейса:

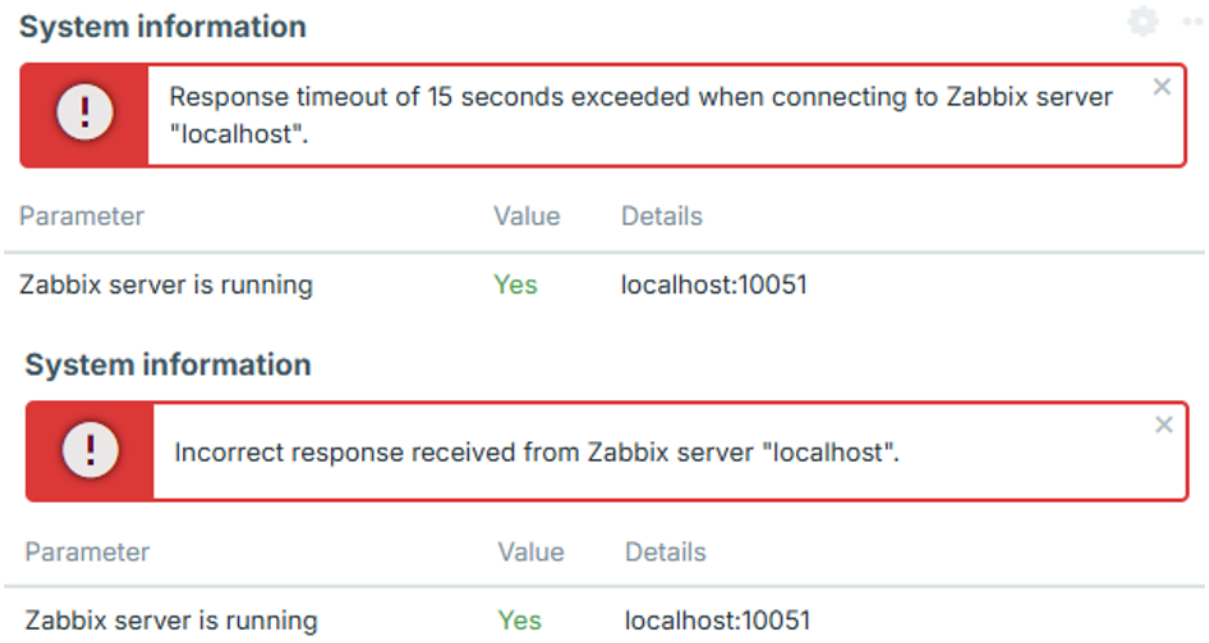
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
//        :
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

5. Проверьте шифрование, убедившись в отсутствии сообщений об ошибках в веб-интерфейсе Zabbix или файле журнала сервера Zabbix:

```
tail -f /var/log/zabbix/zabbix_server.log
```

Устранение неполадок

Неправильная или неполная конфигурация TLS может привести к ошибкам в веб-интерфейсе Zabbix или указывать на то, что сервер Zabbix запущен, но подключение не устанавливается успешно:



Чтобы определить точную проблему конфигурации, установите уровень журнала сервера Zabbix равным 4, воспроизведите ошибку в веб-интерфейсе и проверьте журнал сервера Zabbix на наличие соответствующего сообщения об ошибке.

В следующих случаях описаны распространенные ошибки и их возможные причины.

IP-адрес веб-интерфейса не разрешен

В журнале сервера Zabbix содержится ошибка, подобная следующей:

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

Соединение разрывается после установления TCP-соединения, поскольку IP-адрес веб-интерфейса не авторизован.

Добавьте IP-адрес веб-интерфейса в параметр `FrontendAllowedIP` файла `zabbix_server.conf` и перезапустите сервер Zabbix.

Незашифрованное подключение не разрешено

В журнале сервера Zabbix содержится ошибка, подобная следующей:

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Сервер Zabbix настроен на прием только TLS-подключений, например с помощью `TLSFrontendAccept=cert`, однако веб-интерфейс пытается подключиться без шифрования.

Включите шифрование веб-интерфейса и настройте необходимые файлы TLS в `zabbix.conf.php`, как описано в шаге 4 раздела Конфигурация.

Веб-интерфейс пытается использовать TLS, а сервер ожидает незашифрованное соединение

В журнале сервера Zabbix содержится ошибка, подобная следующей:

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

Точное сообщение об ошибке зависит от версии OpenSSL.

Сервер Zabbix настроен на прием только незашифрованных соединений, например с параметром `TLSFrontendAccept=unencrypte` тогда как веб-интерфейс пытается установить TLS-соединение.

Настройте сервер Zabbix на прием TLS-соединений, установив `TLSFrontendAccept=cert` и задав параметры `TLSCertFile` и `TLSKeyFile` в файле `zabbix_server.conf`. В качестве альтернативы отключите шифрование соединения с веб-интерфейсом в файле `zabbix.conf.php`.

Сертификат веб-интерфейса не является доверенным

В журнале сервера Zabbix содержится ошибка, похожая на следующую:

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

Веб-интерфейс предоставил сертификат, но сервер Zabbix не доверяет центру сертификации (CA), подписавшему этот сертификат.

Та же проблема может отображаться в системной информации веб-интерфейса следующим образом:

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

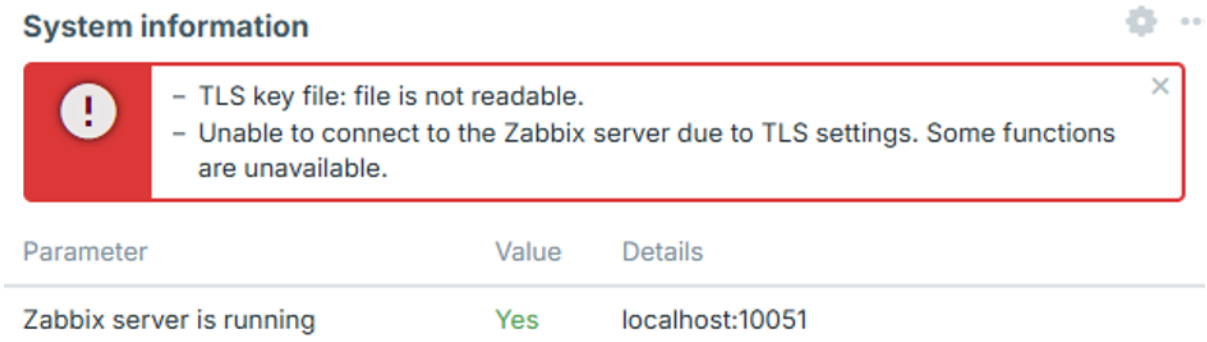
Веб-интерфейс также может отображать сообщение «Получен некорректный ответ от сервера Zabbix "localhost"».

Убедитесь, что сертификат веб-интерфейса подписан центром сертификации, указанным в параметре `TLSCAFile` файла `zabbix_server.conf`.

Файл ключа TLS недоступен для чтения

Веб-интерфейс может отображать сообщение «Файл ключа TLS: файл недоступен для чтения».

Также может отображаться сообщение «Не удалось подключиться к серверу Zabbix из-за настроек TLS. Некоторые функции недоступны».

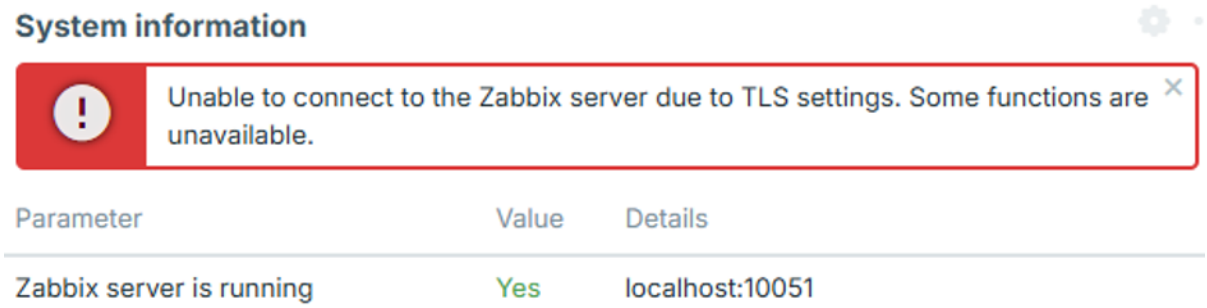


У пользователя веб-сервера недостаточно прав для чтения одного или нескольких файлов сертификатов, ключей или центров сертификации, настроенных в `zabbix.conf.php`. Это может произойти, если файлы сертификатов хранятся в каталоге с ограниченным доступом, например в /root/ или домашнем каталоге пользователя.

Переместите файлы сертификатов в каталог, доступный веб-серверу, и установите соответствующие разрешения, как описано в шаге 2 раздела Настройка.

Файлы TLS недействительны или повреждены

Веб-интерфейс может отображать сообщение: «Невозможно подключиться к серверу Zabbix из-за настроек TLS. Некоторые функции недоступны».



Это может произойти, если настроенные файлы сертификата или ключа существуют и доступны для чтения, но их содержимое пусто, повреждено или иным образом недействительно. В этом случае веб-интерфейс не устанавливает соединение с сервером Zabbix, и в журнале сервера Zabbix не появляется соответствующая попытка подключения.

Убедитесь, что файлы сертификата и ключа содержат корректные данные в правильном формате и что закрытый ключ соответствует сертификату.

Управление доступом

Обзор

В этом разделе приведены рекомендации по безопасной настройке управления доступом.

Принцип наименьших привилегий

Учетные записи пользователей должны всегда работать с минимально возможным набором привилегий. Это означает, что учетные записи пользователей во веб-интерфейсе Zabbix, пользователи базы данных или пользователь для процессов сервера/прокси/агента Zabbix должны иметь только те привилегии, которые необходимы для выполнения предназначенных функций.

Attention:
Предоставление дополнительных привилегий пользователю 'zabbix' позволит ему получать доступ к файлам конфигурации и выполнять операции, которые могут поставить под угрозу безопасность инфраструктуры.

При настройке привилегий учетных записей пользователей следует учитывать типы пользователей веб-интерфейса Zabbix. Обратите внимание, что хотя тип пользователя *Admin* имеет меньше привилегий, чем тип пользователя *Super Admin*, он все равно может управлять конфигурацией и выполнять пользовательские скрипты.

Note:
Некоторая информация доступна даже пользователям без привилегий. Например, хотя *Alerts* → *Scripts* доступно только пользователям *Super Admin*, скрипты также можно получить через API Zabbix. В этом случае ограничение прав на скрипты и исключение конфиденциальной информации из скриптов (например, учетных данных для доступа) может помочь избежать раскрытия конфиденциальной информации, доступной в глобальных скриптах.

Безопасный пользователь для агента Zabbix

По умолчанию процессы сервера, прокси и агента Zabbix (или agent 2) используют одну учетную запись `zabbix`. Чтобы предотвратить доступ агента Zabbix/agent 2 (запущенного на том же компьютере, что и сервер/прокси) к конфиденциальным сведениям в конфигурации сервера/прокси (например, к учетным данным базы данных), агент следует запускать от имени другого пользователя:

Для агента Zabbix:

1. Создайте безопасные группу и пользователя (например, `zabbix-agent`).
2. Укажите этого пользователя в параметре User файла конфигурации агента.
3. Перезапустите агент, чтобы понизить привилегии до нового пользователя.

Для Zabbix agent 2 настройку необходимо применять на уровне службы, поскольку файл конфигурации agent 2 не поддерживает параметр `User`. Пример см. в ZBX-26442.

Отозвать права на запись для конфигурации SSL (Windows)

Если вы скомпилировали агент Zabbix в Windows, а OpenSSL расположен в незащищенном каталоге (например, `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` или `C:\dev\openssl`), обязательно отзовите права на запись в этот каталог у пользователей, не являющихся администраторами. В противном случае агент загружает настройки SSL из пути, который могут изменять пользователи без привилегий, что создает потенциальную уязвимость безопасности.

Усиление безопасности компонентов Zabbix

Некоторые функции можно отключить для усиления безопасности компонентов Zabbix:

- Глобальное выполнение скриптов на сервере Zabbix можно отключить, установив `EnableGlobalScripts=0` в конфигурации сервера.
- Глобальное выполнение скриптов на прокси Zabbix по умолчанию отключено (его можно включить, установив `EnableRemoteCommands=1` в конфигурации прокси).
- Глобальное выполнение скриптов на агентах Zabbix по умолчанию отключено (его можно включить, добавив параметр `AllowKey=system.run[<command>,*]` или `AllowKeyRegexp=^system\.run\[.*\]$` для каждой разрешенной команды в конфигурации агента).
- Доступ к настройке HTTP-аутентификации пользователей по умолчанию включен; его можно отключить, установив `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` в конфигурации веб-интерфейса.
- Доступ к настройке модулей по умолчанию включен; его можно отключить, установив `$ZBX_FEATURE_FLAGS['modules_confi = false` в конфигурации веб-интерфейса.
- Редактирование определенных типов носителей можно ограничить, установив `$ZBX_FEATURE_FLAGS['media_type_denylis = [<value1>, <value2>, ...]` в конфигурации веб-интерфейса. Возможные значения: `'email'`, `'script'`, `'sms'`, `'webhook'` и `'push'`. Можно указать одно или несколько значений.

Переустановка веб-интерфейса путем запуска `setup.php` ограничена, если существует файл конфигурации `zabbix.conf.php`.

Доступ к путям UNC в Windows с помощью агента Zabbix

Агенты Zabbix в Windows обрабатывают пути UNC (SMB-ресурсы, например `\\server\share\file.txt`) в элементах данных, таких как `vfs.file.*`, `vfs.dir.*`, `modbus.get` и `perf_counter*`. В некоторых случаях это может представлять угрозу безопасности.

Когда Windows получает запрос на доступ к пути UNC, она пытается пройти аутентификацию на этом сервере. Это означает, что вредоносный запрос к агенту Zabbix может раскрыть NTLM-хеш серверу, с которого поступил запрос. Пользователи могут снизить этот риск, настроив при необходимости параметры конфигурации AllowKey, AllowKeyRegexp, DenyKey и DenyKeyRegexp.

Защита MySQL/MariaDB

Обзор

В этом разделе приведены рекомендации по обеспечению безопасности базы данных MySQL/MariaDB.

> **Note:**
> Для базовой настройки см. инструкции по умолчанию для создания базы данных MySQL/MariaDB, которые включают создание пользователя 'zabbix' с полными привилегиями в базе данных Zabbix. Этот пользователь является владельцем базы данных и также имеет необходимые привилегии для изменения структуры базы данных при обновлении Zabbix.

Для повышения безопасности рекомендуется создавать дополнительные роли и пользователей базы данных с минимальными привилегиями. Эти роли и пользователи должны быть настроены в соответствии с принципом наименьших привилегий, то есть они должны иметь только те привилегии, которые необходимы для выполнения предполагаемых функций.

Создание ролей пользователей

Создайте следующие роли с соответствующими привилегиями:

- **zbx_srv** - роль для запуска сервера и прокси Zabbix:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - роль для запуска веб-интерфейса и API Zabbix:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - роль для резервного копирования таблиц:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

> **Note:**
> Восстановление таблиц и обновление должны выполняться владельцем базы данных.

- **zbx_part** - роль с сокращенным набором привилегий для секционирования базы данных; обратите внимание, что эту роль можно создать только после создания базы данных, так как она предоставляет привилегии для конкретных таблиц базы данных:

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

После создания роли можно назначать пользователям.

Назначение ролей пользователей

Чтобы назначить созданные роли пользователей, создайте пользователей и назначьте им соответствующие роли. Замените <user>, <host>, <role> и <password> соответствующими значениями. Убедитесь, что значение <password> является надежным и уникальным.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
--    MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Например, чтобы создать и назначить роль для запуска сервера и прокси Zabbix:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

Защита PostgreSQL/TimescaleDB

Обзор

В этом разделе приведены рекомендации по обеспечению безопасности базы данных PostgreSQL.

Note:
Для базовой настройки см. инструкции по умолчанию для создания базы данных PostgreSQL, которые включают создание пользователя 'zabbix' с полными привилегиями в базе данных Zabbix. Этот пользователь является владельцем базы данных и также имеет необходимые привилегии для изменения структуры базы данных при обновлении Zabbix.

Для повышения безопасности рекомендуется настроить безопасный способ использования схемы, а также создать дополнительные роли и пользователей базы данных с минимальными привилегиями. Эти роли и пользователи должны быть настроены в соответствии с принципом наименьших привилегий, то есть они должны иметь только те привилегии, которые необходимы для выполнения предполагаемых функций.

Настройка базы данных

Создайте пользователя, который будет владельцем базы данных, и создайте базу данных Zabbix; владелец базы данных — это пользователь, указанный при создании базы данных:

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

Note:
Чистая установка или обновление базы данных должны выполняться владельцем базы данных. Это связано с тем, что право удалять объект базы данных или изменять его определение является привилегией, присущей владельцу базы данных, и не может быть предоставлено или отозвано.

Attention:
Следующие команды на этой странице должны выполняться при подключении к PostgreSQL именно к базе данных zabbix.

Создайте схему zabbix и назначьте владельца базы данных (usr_owner) владельцем этой схемы:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Настройте безопасный шаблон использования схемы:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
--       : search_path                "zabbix":
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

После настройки базы данных перейдите к созданию ролей пользователей.

Создание ролей пользователей

Создайте следующие роли с соответствующими привилегиями:

- **zbx_srv** - роль для запуска Zabbix server и proxy:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - роль для запуска веб-интерфейса Zabbix и API:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - роль для резервного копирования таблиц:

```
CREATE ROLE zbx_bckp;
GRANT CONNECT ON DATABASE zabbix TO zbx_bckp;
GRANT USAGE ON SCHEMA zabbix TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT ON TABLES TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

Note:
Восстановление таблиц возможно только владельцем базы данных.

После создания ролей их можно назначить пользователям.

Назначение ролей пользователей

Чтобы назначить созданные роли пользователей, создайте пользователей и назначьте им соответствующие роли. Замените <user>, <role> и <password> соответствующими значениями. Убедитесь, что <password> является надежным и уникальным.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Например, чтобы создать роль и назначить ее для запуска сервера и прокси Zabbix:

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-password';
GRANT zbx_srv TO usr_srv;
```

Разделение базы данных с помощью TimescaleDB

Разделение базы данных упрощается благодаря TimescaleDB. Для использования TimescaleDB серверу Zabbix требуются права владельца базы данных.

Если схема PostgreSQL zabbix уже создана в базе данных zabbix, вы можете включить TimescaleDB с помощью следующей команды:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

Криптография

Обзор

В этом разделе приведены рекомендации по безопасной настройке криптографии.

Настройка SSL для веб-интерфейса Zabbix

В системах на базе RHEL установите пакет mod_ssl:

```
dnf install mod_ssl
```

Создайте каталог для SSL-ключей:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Создайте SSL-сертификат:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Корректно заполните запросы. Самая важная строка — та, которая запрашивает `Common Name`. Вы должны указать доменное имя, которое хотите связать с вашим сервером. Если доменного имени нет, вместо него можно указать публичный IP-адрес.

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Отредактируйте файл конфигурации Apache SSL (`/etc/httpd/conf.d/ssl.conf`):

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

Перезапустите службу Apache, чтобы применить изменения:

```
systemctl restart httpd.service
```

Защита веб-сервера

Обзор

В этом разделе приведены рекомендации по безопасной настройке веб-сервера.

Принудительное перенаправление корневого URL на Zabbix SSL

В системах на базе RHEL добавьте виртуальный хост в конфигурацию Apache (`/etc/httpd/conf/httpd.conf`) и задайте постоянное перенаправление для корня документа на URL Zabbix SSL. Обратите внимание, что `example.com` следует заменить на фактическое имя сервера.

```
###         :

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Перезапустите службу Apache, чтобы применить изменения:

```
systemctl restart httpd.service
```

Включение HTTP Strict Transport Security (HSTS) на веб-сервере

Чтобы защитить веб-интерфейс Zabbix от атак понижения версии протокола, мы рекомендуем включить политику HSTS на веб-сервере.

Чтобы включить политику HSTS для вашего веб-интерфейса Zabbix в конфигурации Apache, выполните следующие шаги:

1. Найдите файл конфигурации вашего виртуального хоста:
   - `/etc/httpd/conf/httpd.conf` в системах на базе RHEL
   - `/etc/apache2/sites-available/000-default.conf` в Debian/Ubuntu
2. Добавьте следующую директиву в файл конфигурации вашего виртуального хоста:

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Перезапустите службу Apache, чтобы применить изменения:

```
###             RHEL:
systemctl restart httpd.service

###   Debian/Ubuntu
systemctl restart apache2.service
```

Обеспечение безопасных и SameSite cookie сеанса в Zabbix

При настройке Zabbix важно принудительно задавать атрибуты secure и SameSite для cookie сеанса, чтобы повысить безопасность и предотвратить атаки типа cross-site request forgery (CSRF). Однако принудительное использование `SameSite=Strict` может вызывать проблемы в некоторых сценариях, например:

- Виджеты URL на панели мониторинга отображают сообщение "user not logged in" при встраивании iframe с тем же доменом.
- Пользователи, обращающиеся к панели мониторинга по HTTP вместо HTTPS, могут столкнуться с проблемами входа.
- Невозможно делиться URL на определенные разделы меню Zabbix или узлы сети.

Чтобы смягчить эти проблемы, пользователям должен быть доступен способ настройки политики SameSite.

1. Безопасные cookie

Установка флага `secure` гарантирует, что cookie передаются только по HTTPS, предотвращая их раскрытие через незашифрованные соединения.

Чтобы включить безопасные cookie в Zabbix, добавьте или измените следующую настройку в конфигурации веб-сервера:

Для Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Для Nginx:

```
proxy_cookie_path / "/; Secure";
```

Убедитесь, что веб-интерфейс Zabbix открывается по HTTPS; в противном случае cookie с флагом `Secure` не будут отправляться.

2. Настройка атрибута SameSite

Параметры веб-сервера также могут принудительно задавать атрибут SameSite:

Для Apache:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Для Nginx (версия 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Replace ~ with 'zbx_session' for specificity
```

Включение Content Security Policy (CSP) на веб-сервере

Чтобы защитить веб-интерфейс Zabbix от Cross Site Scripting (XSS), внедрения данных и подобных атак, рекомендуется включить Content Security Policy на веб-сервере. Для этого настройте веб-сервер так, чтобы он возвращал HTTP-заголовок.

> **Attention:**
> Следующая конфигурация заголовка CSP предназначена только для стандартной установки веб-интерфейса Zabbix и для случаев, когда весь контент поступает из домена сайта (за исключением поддоменов). Может потребоваться другая конфигурация заголовка CSP, если вы, например, настраиваете виджет *URL* для отображения контента из поддоменов сайта или внешних доменов, переключаетесь с *OpenStreetMap* на другой картографический движок или добавляете внешний CSS или виджеты. Если вы используете метод многофакторной аутентификации Duo Universal Prompt multi-factor authentication, обязательно добавьте "duo.com" в директиву CSP в файле конфигурации вашего виртуального хоста.

Чтобы включить CSP для вашего веб-интерфейса Zabbix в конфигурации Apache, выполните следующие действия:

1. Найдите файл конфигурации вашего виртуального хоста:

- `/etc/httpd/conf/httpd.conf` в системах на базе RHEL
- `/etc/apache2/sites-available/000-default.conf` в Debian/Ubuntu

2. Добавьте следующую директиву в файл конфигурации вашего виртуального хоста:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Перезапустите службу Apache, чтобы применить изменения:

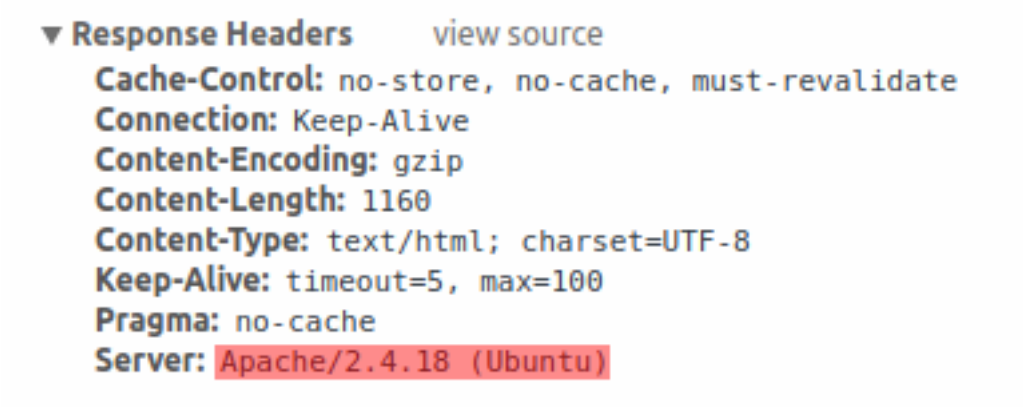
```
###             RHEL:
systemctl restart httpd.service

###   Debian/Ubuntu
systemctl restart apache2.service
```

Отключение раскрытия информации о веб-сервере

Для повышения безопасности рекомендуется отключить все сигнатуры веб-сервера.

По умолчанию веб-сервер раскрывает сигнатуру программного обеспечения:



Сигнатуру можно отключить, добавив следующие параметры в файл конфигурации Apache:

```
ServerSignature Off
ServerTokens Prod
```

Сигнатуру PHP (HTTP-заголовок X-Powered-By) можно отключить, изменив файл конфигурации `php.ini` (по умолчанию сигнатура отключена):

```
expose_php = Off
```

Для применения изменений в файле конфигурации требуется перезапустить веб-сервер.

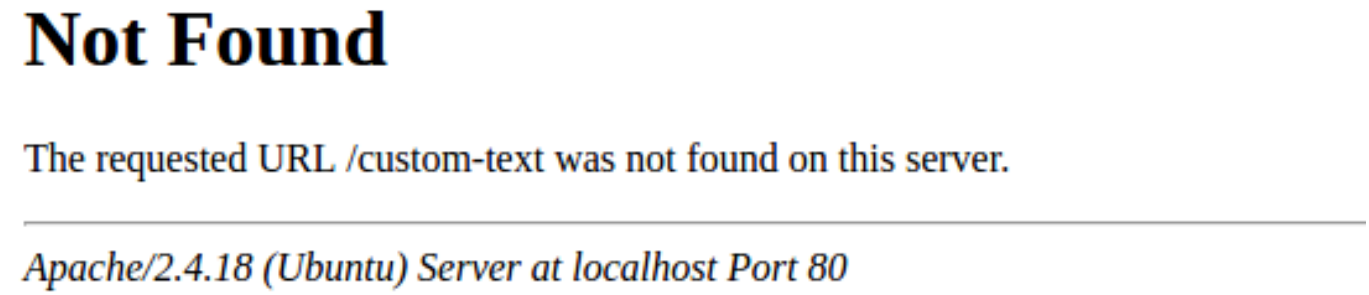
Для дополнительной защиты можно использовать инструмент *mod_security* с Apache (пакет *libapache2-mod-security2*). Этот инструмент позволяет удалить сигнатуру сервера, а не только версию из сигнатуры сервера. Сигнатуру сервера можно изменить на любое значение, задав параметр ”SecServerSignature” после установки *mod_security*.

Обратитесь к документации используемого веб-сервера, чтобы узнать, как удалить или изменить сигнатуры программного обеспечения.

Отключение стандартных страниц ошибок веб-сервера

Чтобы избежать раскрытия информации, рекомендуется отключить стандартные страницы ошибок.

По умолчанию веб-сервер использует встроенные страницы ошибок:



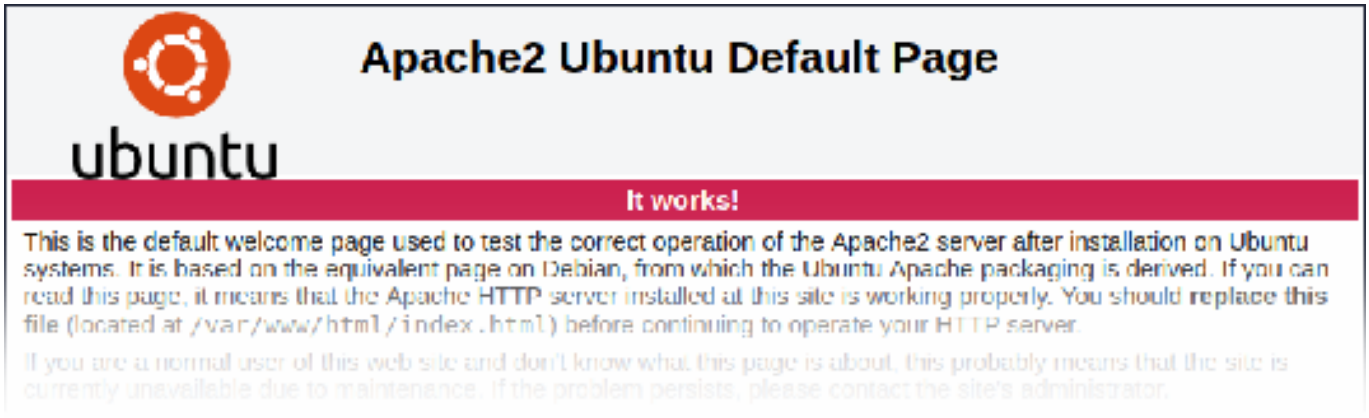
Эти стандартные страницы ошибок следует заменить или удалить. Например, для веб-сервера Apache можно использовать директиву "ErrorDocument", чтобы определить пользовательскую страницу или текст ошибки.

Инструкции по замене или удалению стандартных страниц ошибок см. в документации к используемому веб-серверу.

Удаление тестовой страницы веб-сервера

Для предотвращения раскрытия информации рекомендуется удалить тестовую страницу веб-сервера.

По умолчанию веб-корень веб-сервера Apache содержит тестовую страницу `index.html`:



Обратитесь к документации используемого веб-сервера, чтобы узнать, как удалить тестовые страницы по умолчанию.

Установите HTTP-заголовок ответа X-Frame-Options

По умолчанию Zabbix настроен с параметром *Use X-Frame-Options HTTP header*, установленным в `SAMEORIGIN`. Это означает, что содержимое может быть загружено только в фрейм с тем же origin, что и сама страница.

Элементы веб-интерфейса Zabbix, которые загружают содержимое из внешних URL (а именно, URL виджета панели), отображают полученное содержимое в песочнице со всеми включенными ограничениями песочницы.

Эти настройки повышают безопасность веб-интерфейса Zabbix и обеспечивают защиту от атак XSS и clickjacking. Пользователи с ролью *Super admin* могут изменять параметры *Use iframe sandboxing* и *Use X-Frame-Options HTTP header* по мере необходимости. Пожалуйста, внимательно взвесьте риски и преимущества перед изменением настроек по умолчанию. Полностью отключать iframe sandboxing или HTTP-заголовок X-Frame-Options не рекомендуется.

Скрытие файла со списком распространенных паролей

Чтобы повысить сложность атак перебором пароля, рекомендуется ограничить доступ к файлу `ui/data/top_passwords.txt`. Этот файл содержит список наиболее распространенных и контекстно-зависимых паролей и не позволяет пользователям задавать такие пароли (если включен параметр *Избегать легко угадываемых паролей* в политике паролей).

Чтобы ограничить доступ к файлу `top_passwords.txt`, измените конфигурацию вашего веб-сервера.

В Apache доступ к файлу можно ограничить с помощью файла `.htaccess`:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

В NGINX доступ к файлу можно ограничить с помощью директивы `location`:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

Хранение секретов

Обзор Zabbix можно настроить для получения конфиденциальной информации из защищенного хранилища. Поддерживаются следующие службы управления секретами: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Секреты можно использовать для получения:

- значений пользовательских макросов
- учетных данных для доступа к базе данных

Zabbix предоставляет доступ только для чтения к секретам в хранилище, предполагая, что управление секретами осуществляется кем-то другим.

Для получения информации о конфигурации конкретного поставщика хранилища см.:

- Конфигурация HashiCorp
- Конфигурация CyberArk

Кэширование секретных значений По умолчанию значения макросов секретов из хранилища извлекаются сервером Zabbix при каждом обновлении данных конфигурации, а затем сохраняются в кэше конфигурации. Прокси Zabbix получает значения макросов секретов из хранилища от сервера Zabbix при каждой синхронизации конфигурации и сохраняет их в собственном кэше конфигурации.

> **Attention:**
> Между сервером Zabbix и прокси должно быть включено шифрование; в противном случае в журнал записывается предупреждение сервера.

Также можно настроить независимое извлечение значений макросов сервером Zabbix и прокси Zabbix.

Чтобы вручную запустить обновление кэшированных секретных значений из хранилища, используйте параметр командной строки «secrets_reload».

Кэширование учетных данных базы данных веб-интерфейса Zabbix по умолчанию отключено, но его можно включить, установив параметр `$DB['VAULT_CACHE'] = true` в файле конфигурации веб-интерфейса. Учетные данные будут храниться в локальном кэше с использованием каталога временных файлов файловой системы. Веб-сервер должен разрешать запись в закрытую временную папку (например, для Apache необходимо установить параметр конфигурации `PrivateTmp=True`). Чтобы управлять частотой обновления или аннулирования кэша данных, используйте константу `ZBX_DATA_CACHE_TTL`.

Настройка TLS Чтобы настроить TLS для связи между компонентами Zabbix и хранилищем, добавьте сертификат, подписанный центром сертификации (CA), в системное хранилище CA по умолчанию.
Чтобы использовать другое расположение, укажите каталог в параметре конфигурации Zabbix SSLCALocation server/proxy, поместите файл сертификата в этот каталог, затем выполните команду:

```
c_rehash .
```

Конфигурация CyberArk

В этом разделе объясняется, как настроить Zabbix для получения секретов из CyberArk Vault CV12.

Хранилище должно быть установлено и настроено в соответствии с официальной документацией CyberArk.

Чтобы узнать о настройке TLS в Zabbix, см. *Хранение секретов*.

Учетные данные базы данных

Доступ к секрету с учетными данными базы данных настраивается отдельно для каждого компонента Zabbix.

Сервер и прокси

Чтобы получить учетные данные базы данных из хранилища для сервера Zabbix или прокси, укажите следующие параметры конфигурации в файле конфигурации:

- `Vault` - поставщик хранилища, который следует использовать;
- `VaultURL` - URL-адрес HTTP(S) сервера хранилища;
- `VaultDBPath` - запрос к секрету хранилища, содержащему учетные данные базы данных, которые будут получены по ключам "Content" и "UserName" (этот параметр можно использовать только в том случае, если DBUser и DBPassword не указаны);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - имена файлов SSL-сертификата и ключа; настройка этих параметров не обязательна, но настоятельно рекомендуется;
- `VaultPrefix` - пользовательский префикс для пути или запроса к хранилищу в зависимости от хранилища; если не указан, будет использовано наиболее подходящее значение по умолчанию.

Attention:
Параметры конфигурации `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` и `VaultPrefix` также используются для аутентификации в хранилище при обработке макросов секретов хранилища сервером Zabbix (и прокси Zabbix, если настроено). Сервер Zabbix и прокси не будут раскрывать макросы секретов хранилища, содержащие учетные данные базы данных из VaultDBPath.

Сервер Zabbix и прокси Zabbix считывают параметры конфигурации, связанные с хранилищем, из файлов `zabbix_server.conf` и `zabbix_proxy.conf` при запуске.

Пример

1. В `zabbix_server.conf` укажите следующие параметры:

```
Vault=CyberArk
VaultURL=https://127.0.0.1:1858
VaultDBPath=AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
VaultTLSCertFile=cert.pem
VaultTLSKeyFile=key.pem
VaultPrefix=/AIMWebService/api/Accounts?
```

2. Zabbix отправит в хранилище следующий API-запрос:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. Ответ хранилища будет содержать ключи "Content" и "UserName":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

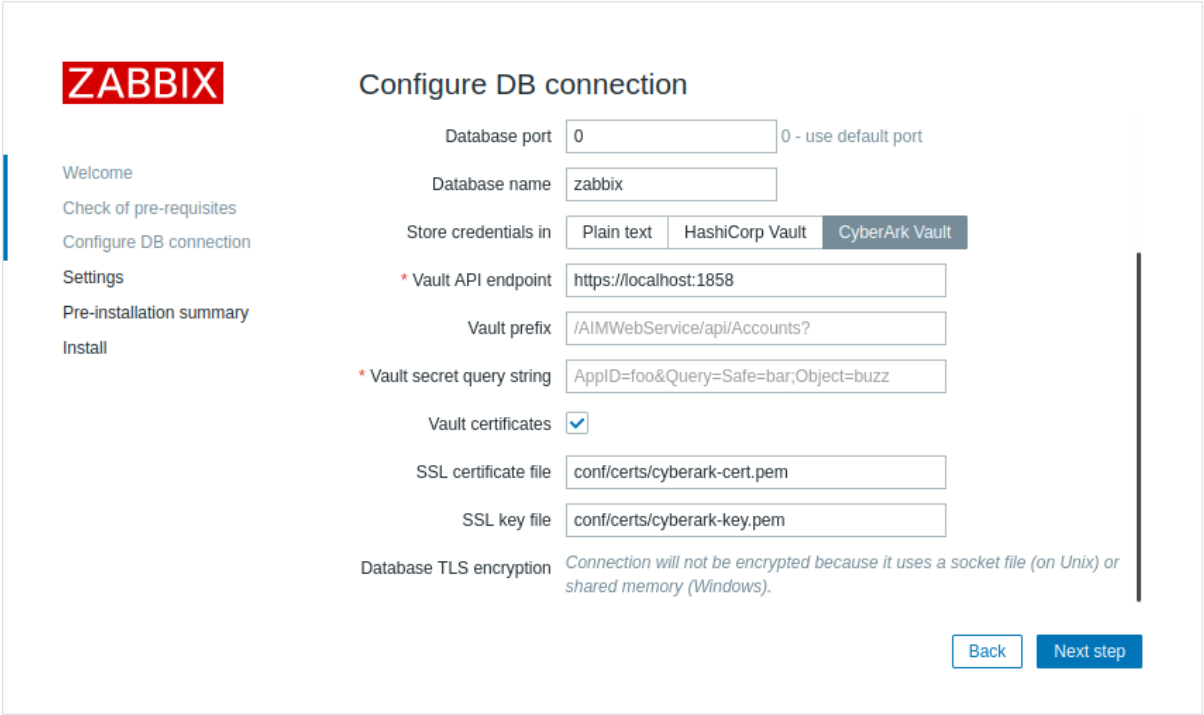
4. В результате Zabbix будет использовать следующие учетные данные для аутентификации в базе данных:

- Имя пользователя: <username>
- Пароль: <password>

Веб-интерфейс

Чтобы получить учетные данные базы данных из хранилища для веб-интерфейса Zabbix, укажите следующие параметры во время установки веб-интерфейса.

1. На шаге *Configure DB Connection* установите для параметра *Store credentials in* значение *CyberArk Vault*.



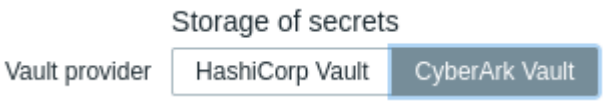
2. Затем заполните дополнительные параметры:

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	Укажите URL для подключения к хранилищу в формате `scheme://host:port`
Vault prefix	no	/AIMWebService/api/Accounts?	Укажите пользовательский префикс для пути или запроса к хранилищу. Если параметр не указан, используется значение по умолчанию.
Vault secret query string	yes		Запрос, указывающий, откуда следует получить учетные данные базы данных. Пример: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		После установки флажка появятся дополнительные параметры, позволяющие настроить аутентификацию клиента. Хотя этот параметр необязателен, настоятельно рекомендуется включить его для обмена данными с CyberArk Vault.
SSL certificate file	no	conf/certs/cyberark-cert.pem	Путь к файлу сертификата SSL. Файл должен быть в формате PEM. Если файл сертификата также содержит закрытый ключ, оставьте параметр SSL key file пустым.
SSL key file	no	conf/certs/cyberark-key.pem	Имя файла закрытого ключа SSL, используемого для аутентификации клиента. Файл должен быть в формате PEM.

Значения пользовательских макросов

Чтобы использовать CyberArk Vault для хранения значений пользовательских макросов *Vault secret*, убедитесь, что:

- сервер Zabbix настроен для работы с CyberArk Vault;
- параметр *Vault provider* в *Administration > General > Other* установлен в "CyberArk Vault".



Note:
Сервер Zabbix (и прокси Zabbix, если настроен) требует доступа к значениям макросов *Vault secret* из хранилища. Веб-интерфейс Zabbix не нуждается в таком доступе.

Значение макроса должно содержать запрос (в формате `query:key`).

См. *Vault secret macros* для подробной информации о том, как Zabbix обрабатывает значение макроса.

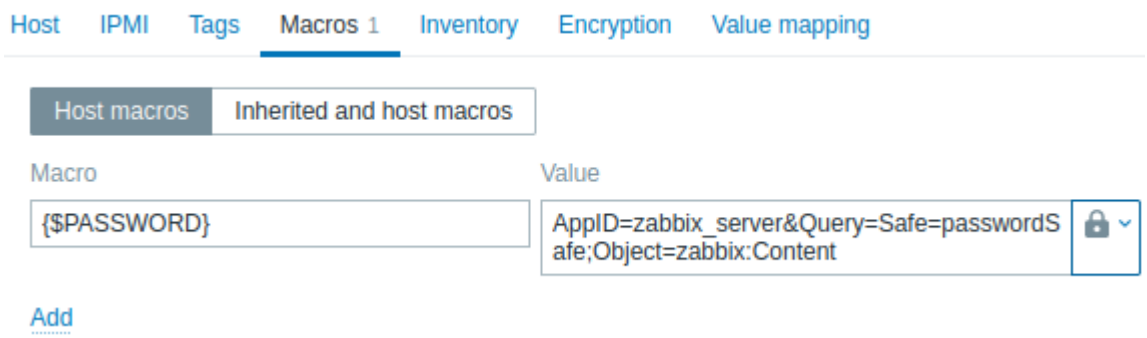
Синтаксис запроса

Символ двоеточия (`:`) зарезервирован для отделения запроса от ключа.

Если сам запрос содержит косую черту или двоеточие, эти символы необходимо закодировать в URL (`/` кодируется как `%2F`, `:` кодируется как `%3A`).

Пример

1. В Zabbix добавьте пользовательский макрос `{$PASSWORD}` типа *Секрет Vault* со значением `AppID=zabbix_server&Query=Saf`



2. Zabbix отправит в хранилище следующий запрос API:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. Ответ хранилища будет содержать ключ `Content`:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. В результате Zabbix подставит в макрос `{$PASSWORD}` значение `<password>`

Обновление существующей конфигурации

Чтобы обновить существующую конфигурацию для получения секретов из CyberArk Vault:

1. Обновите параметры конфигурационного файла сервера или прокси Zabbix, как описано в разделе *Учетные данные базы данных*.

2. Обновите параметры конфигурационного файла веб-интерфейса Zabbix:

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. При необходимости настройте пользовательские макросы, как описано в разделе *Значения пользовательских макросов*.

Чтобы обновить существующую конфигурацию для получения секретов из HashiCorp Vault, см. раздел *Конфигурация HashiCorp*.

Конфигурация HashiCorp

Обзор

В этом разделе объясняется, как настроить Zabbix для получения секретов из HashiCorp Vault KV Secrets Engine - Version 2.

Vault должен быть развернут и настроен в соответствии с официальной документацией HashiCorp.

Чтобы узнать о настройке TLS в Zabbix, см. *Хранение секретов*.

Получение учетных данных базы данных

Для успешного получения секрета с учетными данными базы данных необходимо настроить оба компонента:

- сервер/прокси Zabbix
- веб-интерфейс Zabbix

Сервер/прокси

Чтобы настроить сервер или прокси Zabbix, укажите следующие параметры конфигурации в файле конфигурации:

- `Vault` - поставщик хранилища, который следует использовать.
- `VaultToken` - токен аутентификации хранилища (подробности см. в файле конфигурации сервера/прокси Zabbix); нельзя использовать вместе с `VaultAppRoleID`.
- `VaultAppRoleID` и `VaultAppSecretID` - идентификатор роли и секретный идентификатор для аутентификации в хранилище (подробности см. в файле конфигурации сервера/прокси Zabbix); нельзя использовать вместе с `VaultToken`.
- `VaultURL` - URL-адрес HTTP(S) сервера хранилища.
- `VaultDBPath` - путь к секрету хранилища, содержащему учетные данные базы данных (этот параметр можно использовать только в том случае, если `DBUser` и `DBPassword` не указаны). Сервер или прокси Zabbix получит учетные данные по ключам "password" и "username".
- `VaultPrefix` - пользовательский префикс для пути или запроса хранилища в зависимости от хранилища; если параметр не указан, будет использовано наиболее подходящее значение по умолчанию. Обратите внимание, что префикс должен начинаться и заканчиваться символом `/`.

Attention:
`Параметры конфигурации Vault, VaultToken (или VaultAppRoleID/VaultAppSecretID), VaultURL и VaultPrefix также используются для аутентификации в хранилище при обработке секретных макросов` хранилища сервером Zabbix (и прокси Zabbix, если он настроен). Сервер Zabbix и прокси не будут открывать секретные макросы хранилища, содержащие учетные данные базы данных из VaultDBPath.

Настоятельно рекомендуется использовать разные токены для разных прокси.

Сервер Zabbix и прокси Zabbix считывают параметры конфигурации, связанные с хранилищем, из файлов `zabbix_server.conf` и `zabbix_proxy.conf` при запуске.

Кроме того, сервер Zabbix и прокси Zabbix один раз при запуске считывают переменную окружения `VAULT_TOKEN` и удаляют ее, чтобы она не была доступна через разветвленные скрипты; если параметры `VaultToken` и `VAULT_TOKEN` содержат значения, это считается ошибкой.

Пример

1. В `zabbix_server.conf` укажите следующие параметры.

Используя VaultToken:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Или, используя VaultAppRoleID:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
```

```
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Выполните следующие команды CLI, чтобы создать необходимый секрет в vault:

```
#### "secret/", ; , "kv-v2".
vault secrets enable -path=secret/ kv-v2

#### username password "secret/" "zabbix/database".
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### , .
vault kv get secret/zabbix/database

#### , Curl; , "data" , "/v1" - ,
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. В результате сервер Zabbix получит следующие учетные данные для аутентификации в базе данных:

- Имя пользователя: zabbix
- Пароль: <password>

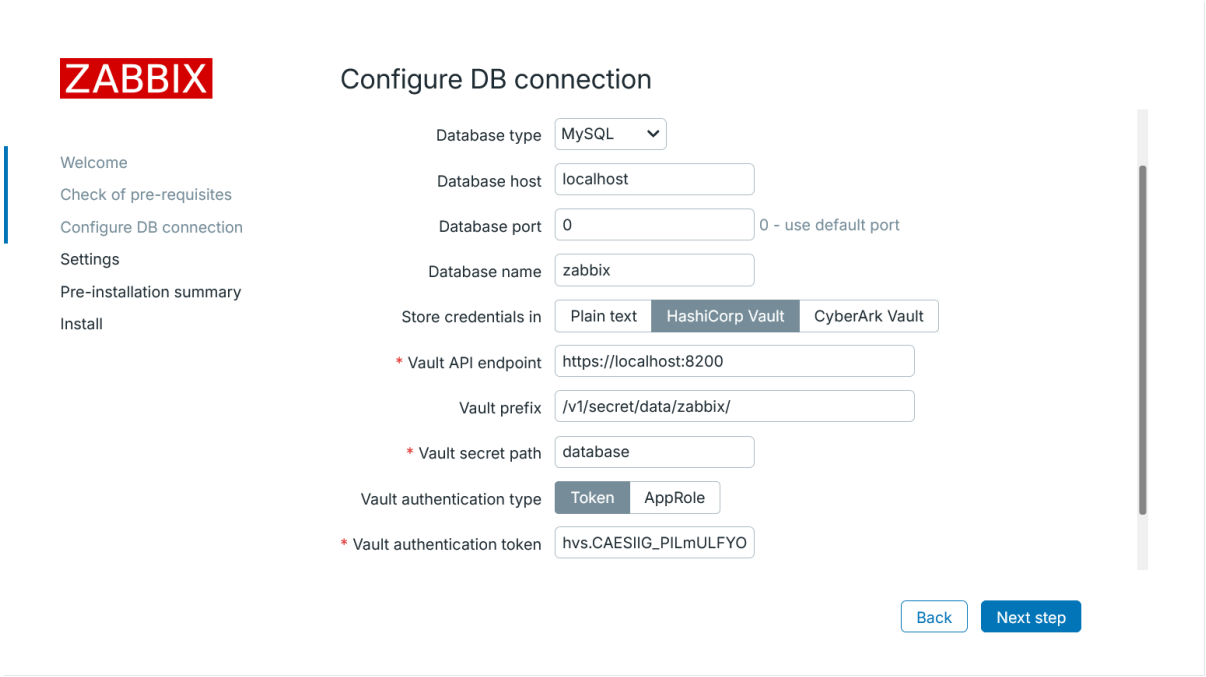
Веб-интерфейс

Веб-интерфейс Zabbix можно настроить для получения учетных данных базы данных из хранилища либо во время установки веб-интерфейса, либо путем обновления файла конфигурации веб-интерфейса (`zabbix.conf.php`).

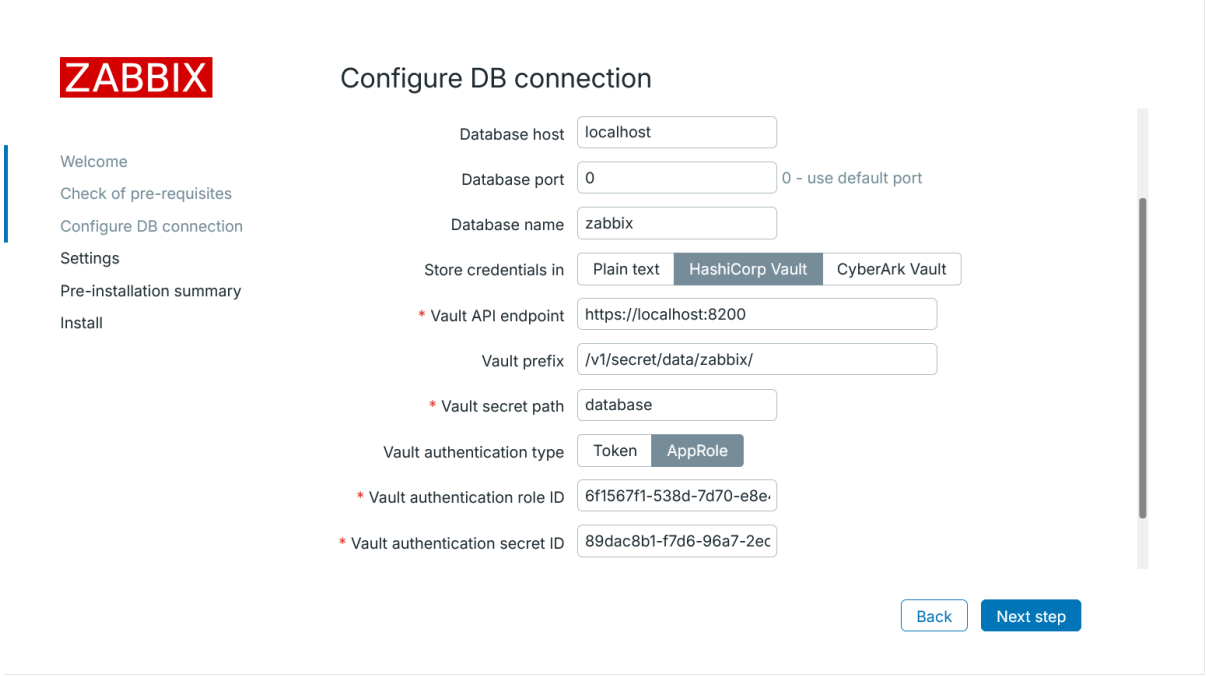
Attention:
Если учетные данные хранилища изменились после предыдущей установки веб-интерфейса, обновите существующую конфигурацию.

Во время **установки веб-интерфейса** параметры подключения необходимо указать на шаге *Configure DB Connection*.

Пример конфигурации с использованием аутентификации по токену:



Пример конфигурации с использованием аутентификации AppRole:



Параметр	Обязательный	Значение по умолчанию	Описание
Store credentials in	нет		Выберите **HashiCorp Vault**.
Vault API endpoint	да	https://localhost:8200	Укажите URL для подключения к хранилищу в формате `scheme://host:port`
Vault prefix	нет	/v1/secret/data/	Укажите пользовательский префикс для пути или запроса хранилища. Если параметр не указан, используется значение по умолчанию. Пример: `/v1/secret/data/zabbix/`
Vault secret path	да		Путь к секрету, из которого будут получены учетные данные базы данных по ключам ”password” и ”username”. Пример: `database`
Vault authentication type	нет		Выберите тип аутентификации хранилища: **Token** - использовать аутентификацию по токену; **AppRole** - использовать метод аутентификации AppRole.
Vault authentication token	да		Укажите токен аутентификации с доступом только для чтения к пути секрета. Информацию о создании токенов и политиках хранилища см. в документации HashiCorp. Это поле доступно только если в качестве типа аутентификации хранилища выбран вариант *Token*.
Vault authentication role ID	да		Укажите идентификатор роли AppRole. Информацию об AppRole см. в документации HashiCorp. Это поле доступно только если в качестве типа аутентификации хранилища выбран вариант *AppRole*.
Vault authentication secret ID	да		Укажите идентификатор секрета, выданный для AppRole. Информацию об AppRole см. в документации HashiCorp. Это поле доступно только если в качестве типа аутентификации хранилища выбран вариант *AppRole*.

Получение значений пользовательских макросов

Чтобы использовать HashiCorp Vault для хранения значений пользовательских макросов *Vault secret*, убедитесь, что:

- Zabbix сервер/прокси настроен для работы с HashiCorp Vault.
- Параметр *Vault provider* в *Administration > General > Other* установлен в значение ”HashiCorp Vault” (по умолчанию).



Note:
Zabbix сервер (и Zabbix прокси, если настроен) требует доступа к значениям макросов *Vault secret* из хранилища. Веб-интерфейс Zabbix не нуждается в таком доступе.

Значение макроса должно содержать ссылочный путь (в формате `path:key`, например, `macros:password`). Токен аутентификации, указанный при настройке Zabbix сервер/прокси (с помощью параметра `VaultToken`), должен предоставлять доступ только для чтения к этому пути.

Подробную информацию о обработке значений макросов Zabbix см. в разделе *Vault secret macros*.

Синтаксис пути

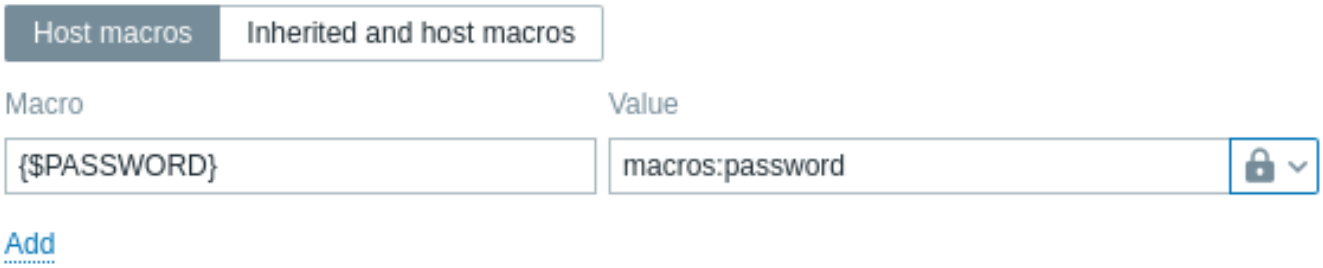
Символы прямой слэш (”/”) и двоеточие (”:”) зарезервированы.

Прямой слэш можно использовать только для разделения точки монтирования и пути (например, *secret/zabbix*, где точка монтирования — ”secret”, а путь — ”zabbix”). В случае макросов Vault двоеточие можно использовать только для разделения пути/запроса и ключа.

При необходимости создать точку монтирования с именем, содержащим прямой слэш (например, *foo/bar/zabbix*, где точка монтирования — ”foo/bar”, а путь — ”zabbix”, можно закодировать как ”foo%2Fbar/zabbix”), а также если имя точки монтирования или путь должны содержать двоеточие, можно использовать URL-кодирование символов прямой слэш и двоеточие.

Пример

1. В Zabbix добавьте пользовательский макрос {$PASSWORD} типа ”Vault secret” со значением `macros:password`



2. Выполните следующие команды CLI, чтобы создать необходимый секрет в vault:

```
####                  "secret/",              ;          ,                "kv-v2".
vault secrets enable -path=secret/ kv-v2

####                    "password"             "secret/"      "zabbix/macros".
vault kv put -mount=secret zabbix/macros password=<password>

####      ,                 .
vault kv get secret/zabbix/macros

####     ,             Curl;          ,    "data"                              "/v1"                ,
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. В результате Zabbix разрешит макрос {$PASSWORD} в значение: <password>

Обновление существующей конфигурации

Чтобы обновить существующую конфигурацию для получения секретов из HashiCorp Vault:

1. Обновите параметры конфигурационного файла Zabbix сервера или прокси, как описано в разделе *Учетные данные базы данных*.

2. Обновите параметры конфигурационного файла веб-интерфейса Zabbix:

С использованием токена Vault:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Или с использованием метода Vault AppRole:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. При необходимости настройте пользовательские макросы, как описано в разделе *Значения пользовательских макросов*.

Чтобы обновить существующую конфигурацию для получения секретов из CyberArk Vault, см. раздел *Конфигурация CyberArk*.

Устранение неполадок

1. В файле журнала сервера/прокси Zabbix:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Вероятная причина: HashiCorp Vault был (пере)запущен, но не был unseal'нут (это требуется в конфигурации по умолчанию, но не требуется, если настроен Auto Unseal).

2. В файле журнала сервера/прокси Zabbix:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

или

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Вероятная причина: VaultAppSecretID истек или достиг лимита использования в HashiCorp vault.

Доверие к TLS-сертификату Google Chrome

Обзор

На этой странице приведены шаги по настройке Zabbix и примеры конфигурации для использования Google Chrome с веб-интерфейсом Zabbix или конечными точками zabbix-web-service, защищенными самоподписанным сертификатом или сертификатом частного центра сертификации.

Эти инструкции предполагают, что целевой веб-сервер уже настроен для работы по HTTPS. Сведения о настройке TLS для веб-интерфейса Zabbix см. в разделе Безопасное подключение к веб-интерфейсу.

Конфигурация

Google Chrome в Linux использует базу данных сертификатов NSS для доверенных сертификатов, привязанную к пользователю. Чтобы Chrome доверял самоподписанному сертификату или частному центру сертификации, добавьте необходимые сертификаты в базу данных NSS учетной записи пользователя, под которой запускается Chrome.

1. Установите необходимые пакеты.

Для Debian/Ubuntu:

```
sudo apt install ca-certificates libnss3-tools
```

Для систем на базе RHEL:

```
sudo dnf install ca-certificates nss-tools
```

Или:

```
sudo yum install ca-certificates nss-tools
```

2. Подготовьте файлы сертификатов.

Для каждого необходимого сертификата: создайте файл `.crt` в формате PEM для самоподписанного сертификата или создайте отдельные файлы `.crt` в формате PEM для сертификата корневого центра сертификации и, если требуется, для каждого промежуточного сертификата.

3. Создайте каталог базы данных NSS.

Используйте домашний каталог пользователя, под которым запускается Chrome.

Например:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Если Chrome в целевой системе использует другое расположение базы данных NSS, используйте этот каталог.

Убедитесь, что файлы сертификатов доступны для чтения пользователю, под которым запускается Google Chrome. При необходимости скопируйте их в место, доступное этому пользователю.

4. Импортируйте сертификаты.

Если используется самоподписанный сертификат:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

Если используется частный центр сертификации, импортируйте в базу данных сертификат корневого центра сертификации:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

Если используются какие-либо промежуточные сертификаты, импортируйте каждый из них отдельно:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. Проверьте базу данных сертификатов.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Проверьте Chrome.

Google Chrome использует базу данных сертификатов учетной записи пользователя, который запускает его. Если у учетной записи пользователя нет доступного для записи домашнего каталога, настройте доступные для записи домашний каталог и каталоги XDG перед выполнением проверки.

Пример (замените `google-chrome` на `chromium`, если используется Chromium.):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Если конфигурация верна, команда вернет HTML запрошенной страницы.

Если Chrome не доверяет сертификату, команда вернет страницу с ошибкой сертификата, содержащую такие сообщения, как `NET::ERR_CERT_AUTHORITY_INVALID`.

Устранение неполадок

Ошибка	Возможная причина
NET::ERR_CERT_AUTHORITY_INVALID	Сертификат был импортирован в неправильный профиль пользователя, отсутствует сертификат корневого центра сертификации или отсутствует промежуточный сертификат.
NET::ERR_CERT_COMMON_NAME_INVALID	Сертификат не соответствует имени хоста, используемому в URL браузера.
NET::ERR_CERT_DATE_INVALID	Срок действия сертификата истек, он еще не вступил в силу или системные часы указаны неверно.
Chrome по-прежнему показывает предупреждение о сертификате после импорта	Chrome использует другую базу данных NSS для текущего пользователя, либо неверно заданы флаги доверия сертификата.

Лучшие практики безопасности

Обзор В этом разделе приведены рекомендации по безопасной настройке Zabbix.

Соблюдение рекомендаций из этого раздела не является обязательным для работы Zabbix, но рекомендуется для повышения безопасности системы.

Рекомендации по обеспечению безопасности мобильного приложения Zabbix и адаптера Zabbix bridge см. в разделе Безопасный мобильный доступ.

Кодировка UTF-8 UTF-8 — единственная кодировка, поддерживаемая Zabbix. Известно, что она работает без каких-либо уязвимостей безопасности. Пользователям следует учитывать, что при использовании некоторых других кодировок известны проблемы безопасности.

Пути установщика Windows При использовании установщиков Windows рекомендуется использовать пути по умолчанию, предлагаемые установщиком. Использование пользовательских путей без соответствующих разрешений может поставить под угрозу безопасность установки.

Макросы в пользовательских глобальных скриптах Для повышения безопасности рекомендуется использовать функции макросов вместо обычных макросов в пользовательских глобальных скриптах, поскольку макросы не экранируются автоматически.

Уведомления о безопасности Zabbix и база данных CVE См. *Уведомления о безопасности Zabbix и база данных CVE*.

Шаблоны HTML-сообщений При создании или редактировании шаблонов сообщений, используемых для отправки сообщений в формате HTML, всегда оборачивайте каждый макрос в функцию макроса `htmlencode()`. Например:

```
<b>          </b>   {{EVENT.TIME}.htmlencode()} {{EVENT.DATE}.htmlencode()}<br><b>          :</b>
{{EVENT.NAME}.htmlencode()}<br><b>      :</b> {{HOST.NAME}.htmlencode()}
<br><b>     :</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>
   :</b> {{EVENT.OPDATA}.htmlencode()}<br><b>      ID      :</b> {{EVENT.ID}.htmlencode()}<br>{{TRIGGER.URL
```

Использование `htmlencode()` гарантирует экранирование любых символов HTML в значениях макросов и предотвращает внедрение HTML-кода в уведомления (например, добавление злоумышленником вредоносной или фишинговой ссылки в уведомление).

Примечание: в стандартных сообщениях электронной почты в формате HTML, предоставляемых Zabbix, функция `htmlencode()` уже применяется к макросам. Эта рекомендация относится к редактированию существующих шаблонов и созданию новых: перед использованием шаблона для отправки сообщения в формате HTML убедитесь, что макросы закодированы.

Веб-интерфейс

Обзор Для удобного доступа к Zabbix из любого места и с любой платформы предоставляется веб-интерфейс.

> **Note:**
> При использовании нескольких экземпляров веб-интерфейса убедитесь, что локали и библиотеки (LDAP, SAML и т. д.) установлены и настроены одинаково для всех веб-интерфейсов.

Справка веб-интерфейса Ссылка на справку ? расположена в правом верхнем углу разделов и форм веб-интерфейса Zabbix и ведёт непосредственно к соответствующим разделам документации.

Для полей формы со значком справки ● при наведении указателя мыши на значок отображается соответствующая подсказка.

Режим киоска Для разделов веб-интерфейса с кнопкой *Режим киоска* (расположенной в правом верхнем углу) нажмите эту кнопку, чтобы переключить раздел в режим киоска. В режиме киоска отображается только содержимое раздела.

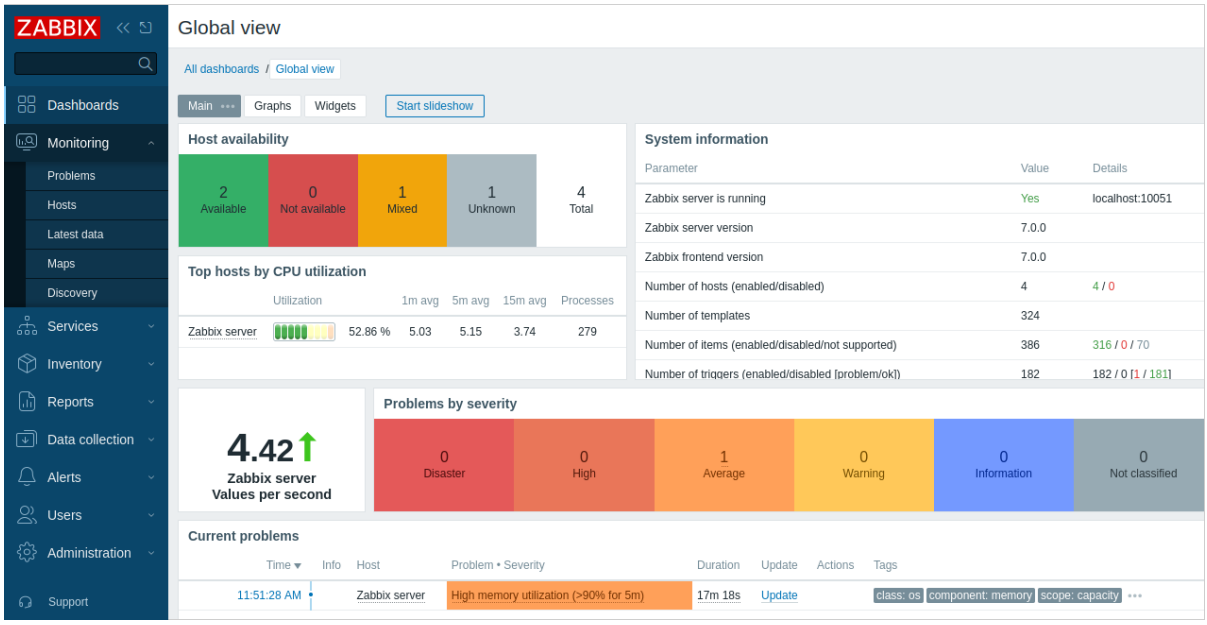
Чтобы выйти из режима киоска, наведите указатель мыши на содержимое и нажмите появившуюся в правом верхнем углу кнопку *Обычный вид*.

Меню

Обзор

Вертикальное меню на боковой панели обеспечивает доступ к различным разделам веб-интерфейса Zabbix.

В теме по умолчанию меню тёмно-синего цвета.

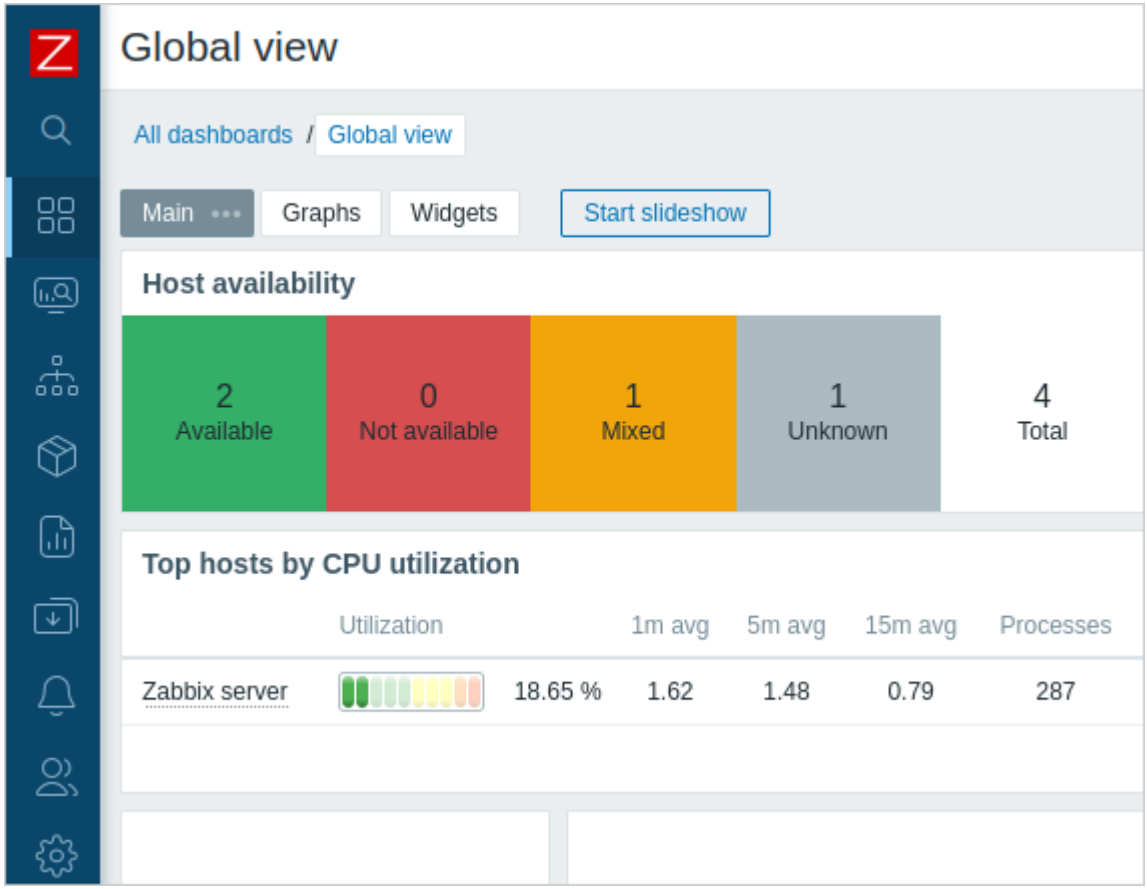


Работа с меню

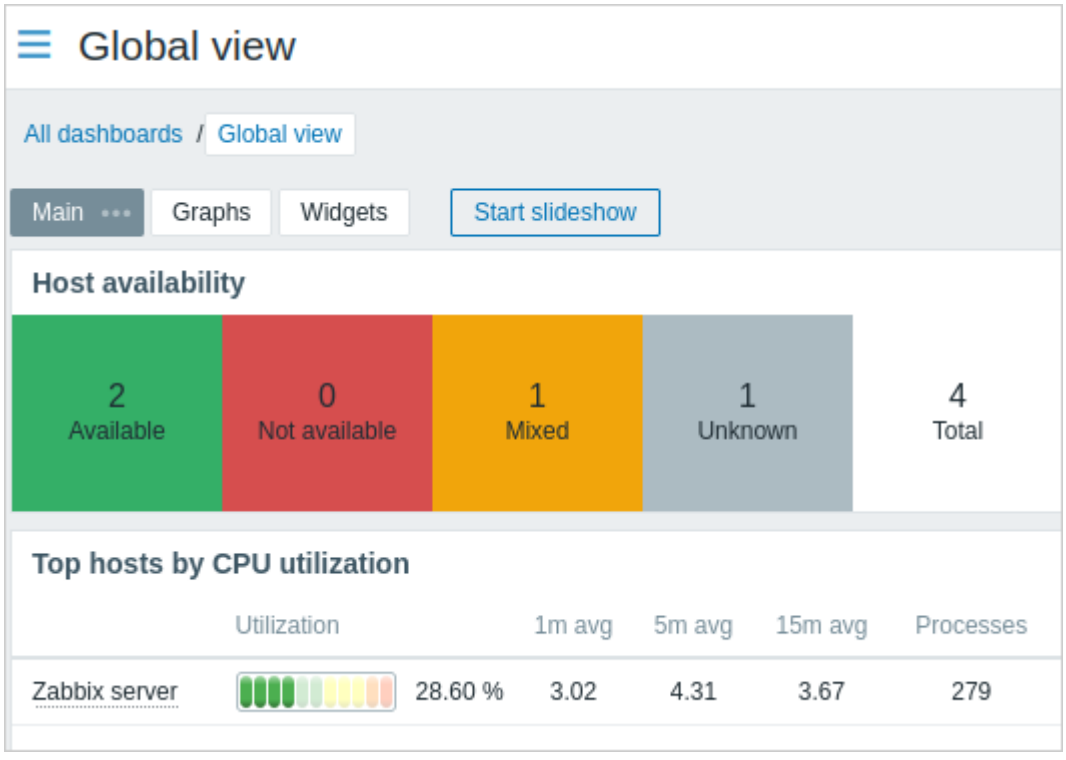
Поле глобального поиска расположено под логотипом Zabbix.

Меню можно свернуть или полностью скрыть:

- Чтобы свернуть меню, нажмите рядом с логотипом Zabbix. В свернутом меню видны только значки.



- Чтобы скрыть меню, нажмите  рядом с логотипом Zabbix. В скрытом меню все элементы скрыты.



Свернутое меню

Если меню свернуто и отображаются только значки, полное меню появляется снова, как только на него наводится курсор мыши. Обратите внимание, что оно появляется поверх содержимого страницы; чтобы переместить содержимое страницы вправо, необходимо нажать кнопку разворачивания. Если снова переместить курсор мыши за пределы полного меню, меню снова свернется через две секунды.

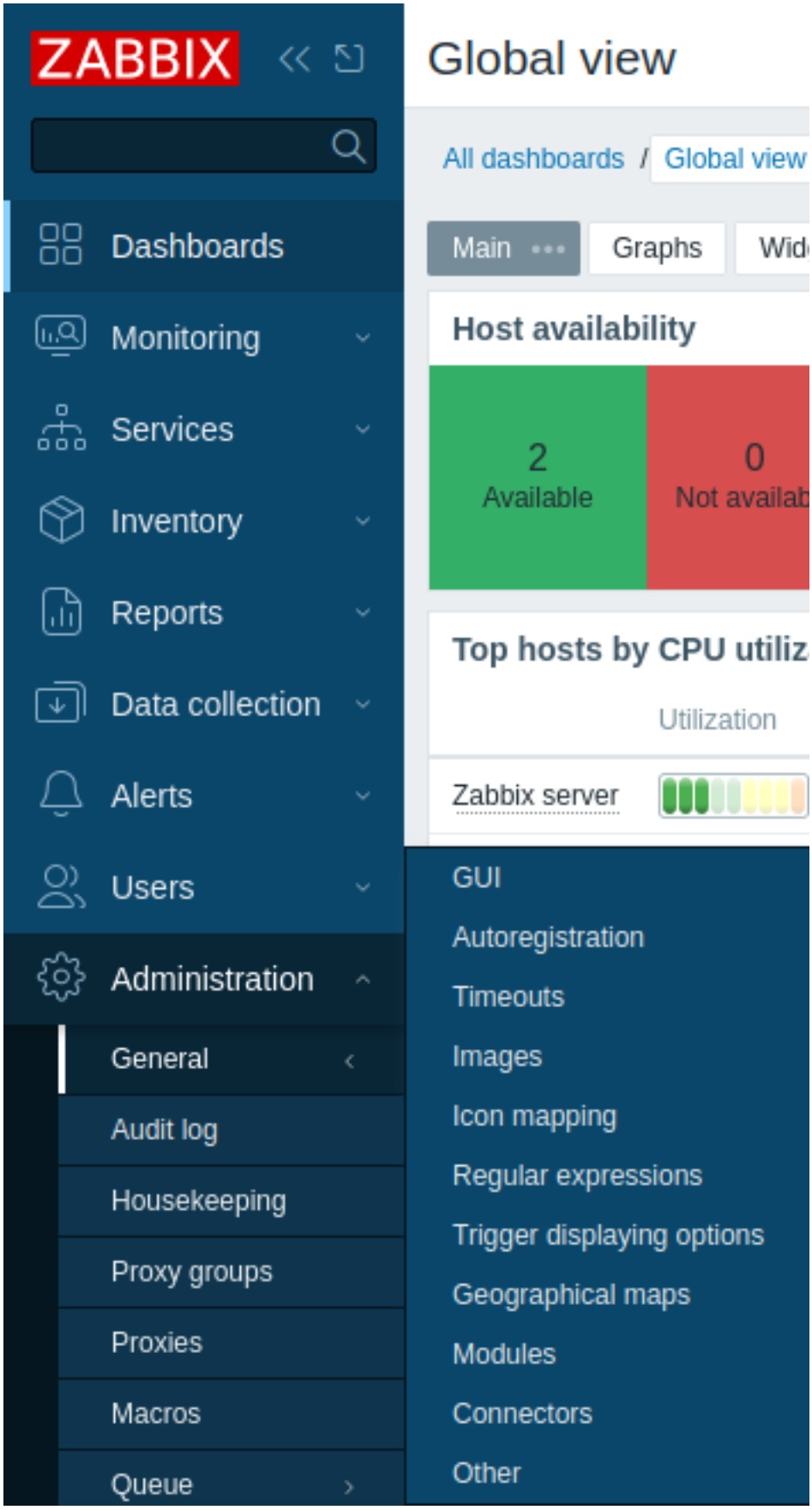
Также можно полностью развернуть свернутое меню, нажав клавишу Tab. Повторное нажатие клавиши Tab позволяет перейти к следующему элементу меню.

Скрытое меню

Даже если меню полностью скрыто, полный список меню можно открыть всего одним щелчком мыши, нажав на значок бургера . Обратите внимание, что оно появляется поверх содержимого страницы; чтобы сдвинуть содержимое страницы вправо, нужно снова показать меню, нажав кнопку отображения боковой панели.

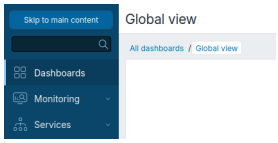
Уровни меню

В меню может быть до трёх уровней.



Доступность: пропустить навигацию

Ссылка *Перейти к основному содержимому* позволяет пользователям клавиатуры и экранных дикторов пропускать повторяющуюся навигацию в боковой панели и сразу переходить к основному содержимому.



В обычном режиме использования она остается скрытой и появляется только при получении фокуса, например когда после загрузки страницы нажимается клавиша Tab или когда к ней обращаются с помощью экранного диктора.

Контекстные меню

Помимо главного меню, Zabbix предоставляет контекстные меню для узел сети, элемент данных и событие для быстрого доступа к часто используемым сущностям, таким как последние значения, простой график, форма настройки, связанные скрипты или внешние ссылки.

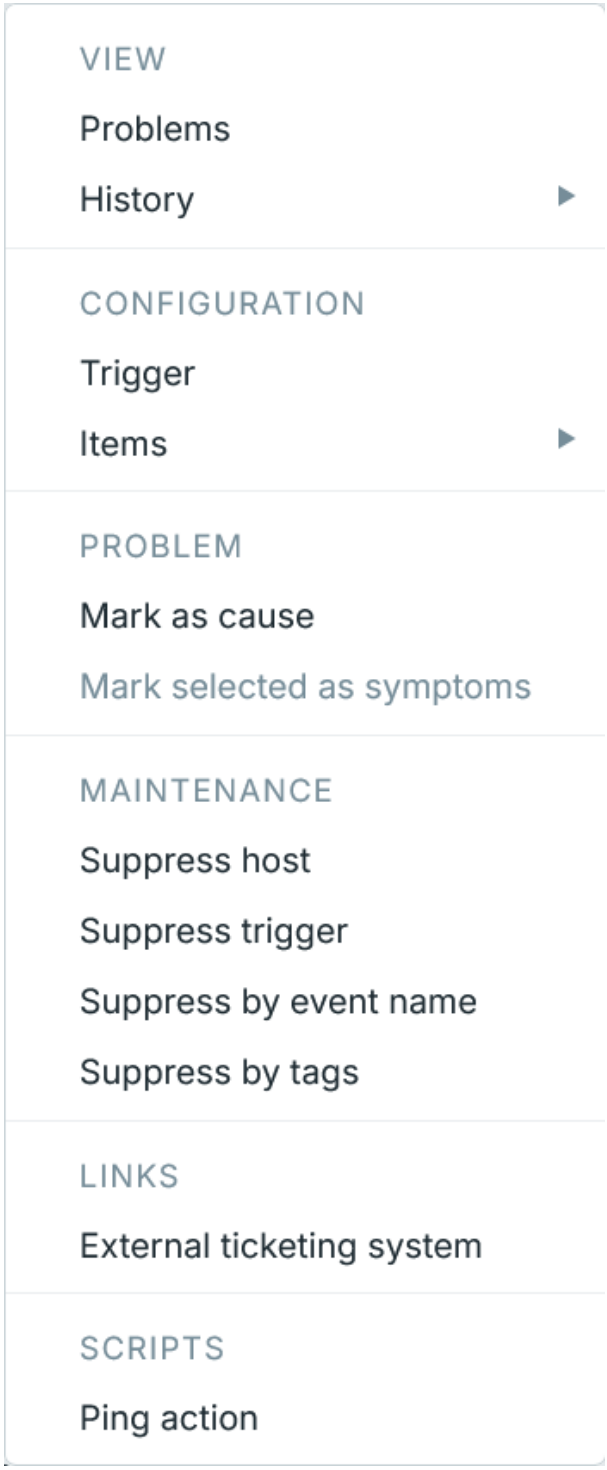
Контекстные меню доступны при нажатии на имя узла сети, элемента данных или проблемы/триггера в поддерживаемых местах.

Меню событий

Обзор

Меню события содержит ссылки для быстрого доступа к действиям или разделам веб-интерфейса, которые часто требуются для события.

Меню события можно открыть, нажав на имя события.



Содержимое

Контекстное меню события содержит несколько разделов: *Просмотр*, *Действия*, *Конфигурация*, *Проблема*, *Обслуживание*, *Ссылки* и *Сценарии*. Для ненастроенных сущностей ссылки отключены и отображаются серым цветом. Разделы *Сценарии* и *Ссылки* отображаются, если соответствующие сущности настроены.

Раздел *Просмотр* содержит ссылки:

- **Проблемы** - открывает список нерешенных проблем базового триггера;
- **История** - открывает график/историю элементов данных в разделе *Последние данные* для базовых элементов данных. Если триггер использует несколько элементов данных, для каждого из них будет доступна отдельная ссылка.

Раздел *Действия* доступен только в виджетах *Обзор триггеров*. Он содержит ссылку:

- **Обновить проблему** - открывает экран обновления проблемы.

Раздел *Конфигурация* содержит ссылки на настройку:

- **Триггера**, вызвавшего проблему;

- **Элементов данных**, используемых в выражении триггера.

Note:
Обратите внимание, что раздел конфигурации доступен только пользователям с ролями «Администратор» и «Суперадминистратор».

Раздел *Проблема* содержит следующие параметры:

- **Отметить как причину** - отметить проблему как причину;
- **Отметить выбранные как симптомы** - отметить выбранные проблемы как симптомы этой проблемы.

Раздел *Обслуживание* содержит параметры для запуска внепланового обслуживания:

- **Подавить узел сети** - запустить период обслуживания для узла сети;
- **Подавить триггер** - запустить период обслуживания для триггера;
- **Подавить по имени события** - запустить период обслуживания для имени события;
- **Подавить по тегам** - запустить период обслуживания для событий с соответствующими тегами.

При выборе соответствующего параметра откроется форма настройки периода обслуживания.

Будет предварительно выбран одноразовый период обслуживания продолжительностью один час (настраивается в настройках пользователя и профиле), начинающийся с текущего момента. Соответствующий узел сети, триггер, имя события или теги также будут предварительно выбраны. Имя обслуживания будет предварительно заполнено в формате `Ad-hoc: < >`.

Раздел *Ссылки* содержит ссылки:

- для доступа к настроенному URL триггера;
- для доступа к пользовательским ссылкам, настроенным в разделе Глобальные сценарии (с областью действия «Ручное действие с событием» и типом «URL»);
- для доступа к настроенной внешней заявке для проблемы (см. параметр *Включить пункт меню события* при настройке вебхуков).

Раздел *Сценарии* содержит ссылки для выполнения глобального сценария (с областью действия *Ручное действие с событием*). Эта функция может быть полезна для запуска сценариев, используемых для управления заявками о проблемах во внешних системах.

Поддерживаемые расположения

Контекстное меню события доступно при нажатии на имя проблемы или события в различных разделах веб-интерфейса, например:

- Панели виджеты, такие как виджет *Problems*, виджет *Trigger overview* и т. д.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Сведения о событии
- *Reports* > *Top 100 triggers* (глобальные скрипты и доступ к внешнему тикету в этом расположении не поддерживаются)

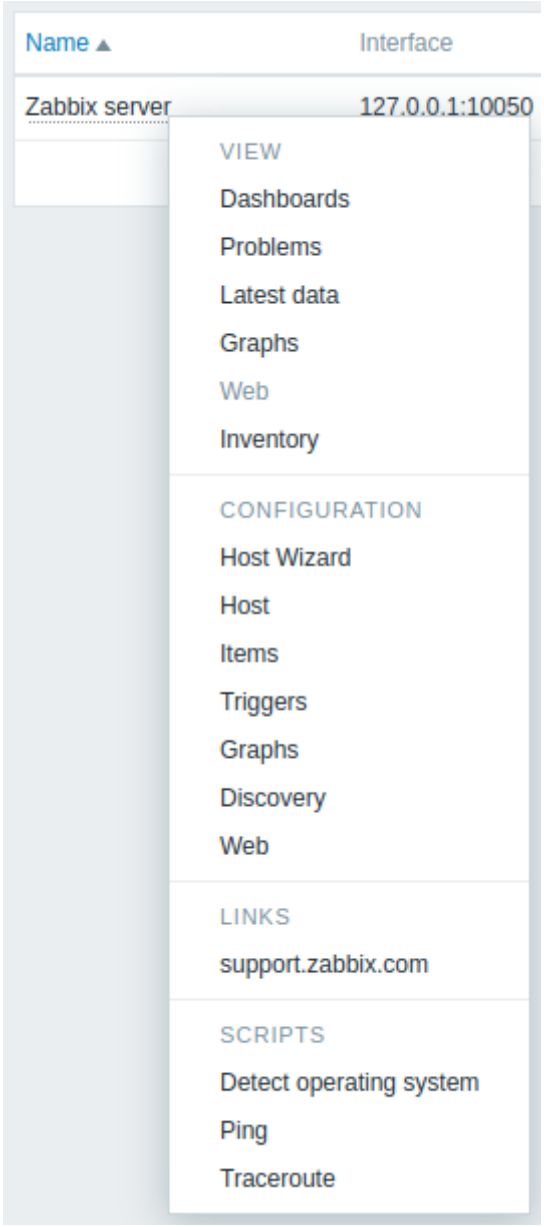
Меню узла сети

Обзор

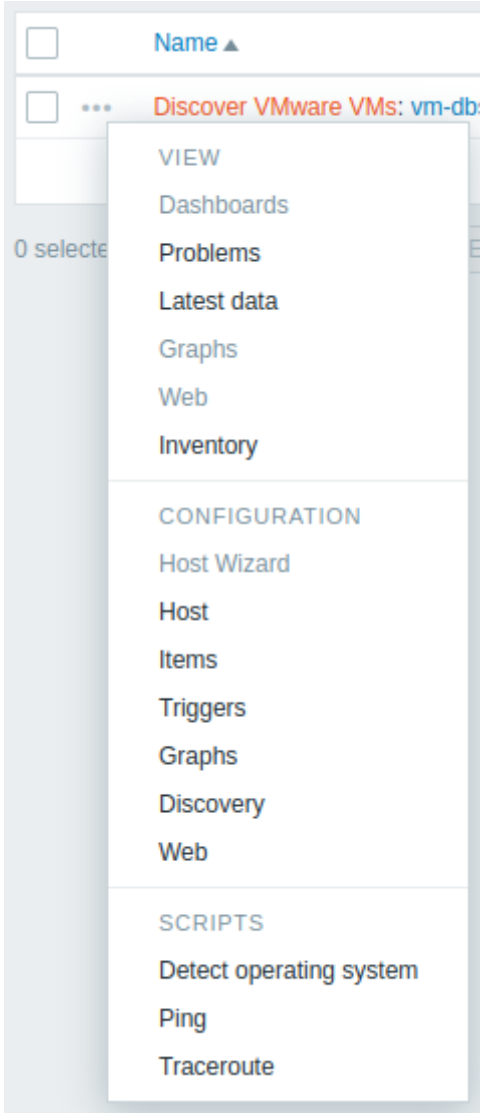
Меню узла сети содержит ярлыки для действий или разделов веб-интерфейса, которые часто требуются для узла сети.

Меню узла сети можно открыть, щелкнув по имени узла сети или значку с тремя точками, в зависимости от раздела веб-интерфейса, например:

Monitoring > *Latest data*



Data collection > *Hosts*



Содержимое

Контекстное меню узла сети содержит четыре раздела: *Просмотр*, *Конфигурация*, *Ссылки* и *Скрипты*. Для ненастроенных сущностей ссылки отключены и отображаются серым цветом. Разделы *Скрипты* и *Ссылки* отображаются, если соответствующие сущности настроены.

Раздел *Просмотр* содержит ссылки на:

- **Панели мониторинга** - открывает виджеты и графики.
- **Проблемы** - открывает раздел *Проблемы* со списком нерешенных проблем, связанных с базовым триггером.
- **Последние данные** - открывает раздел *Последние данные* со списком всех последних данных текущего узла сети.
- **Графики** - открывает простые графики текущего узла сети.
- **Веб** - открывает ссылку на настроенные веб-сценарии.
- **Инвентаризация** - открывает ссылку на инвентаризацию текущего узла сети.

Раздел *Конфигурация* содержит ссылки на:

- **Мастер настройки узла сети** - открывает Мастер настройки узла сети для текущего узла сети (недоступен для обнаруженных узлов сети).
- **Узел сети** - форму настройки текущего узла сети.
- **Элементы данных** - список элементов данных текущего узла сети.
- **Триггеры** - список триггеров текущего узла сети.
- **Графики** - простые графики текущего узла сети.

- **Обнаружение** - список правил обнаружения низкого уровня текущего узла сети.
- **Веб** - список веб-сценариев текущего узла сети.

> **Note:**
> Обратите внимание, что раздел конфигурации доступен только пользователям с правами администратора и супер-администратора.

Раздел *Ссылки* содержит ссылки для:

- доступа к настроенному URL триггера.
- доступа к пользовательским ссылкам, настроенным в разделе Глобальные скрипты (с областью действия *Ручное действие для узла сети* и типом «URL»).

Раздел *Скрипты* позволяет выполнять глобальные скрипты, настроенные для текущего узла сети. Чтобы эти скрипты были доступны в меню узла сети, для них необходимо указать область действия *Ручное действие для узла сети*.

Поддерживаемые расположения

Меню узла сети доступно при нажатии на имя узла сети в различных разделах веб-интерфейса, например:

- Панели виджеты, такие как *Problems*, *Top items*, *Trigger overview* и т. д.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Сведения о событии
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

Меню узла сети также доступно при нажатии на значок с тремя точками в разделе Data collection > Hosts.

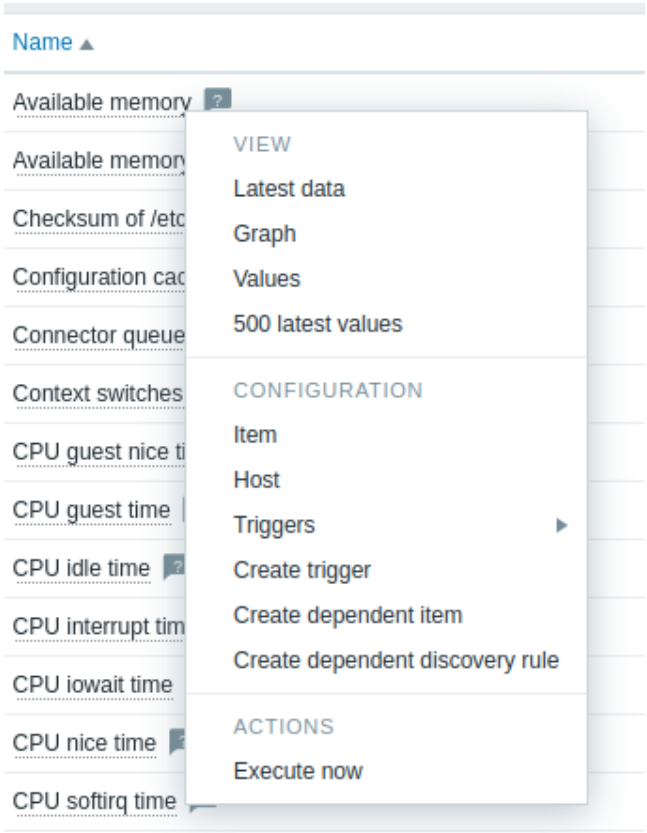
Меню элемента данных

Обзор

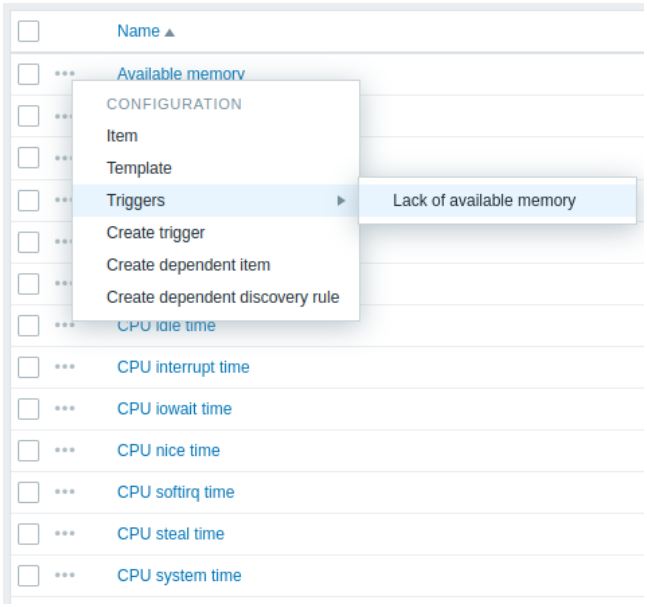
Меню элемента данных содержит ярлыки для действий или разделов веб-интерфейса, которые часто требуются для элемента данных.

Меню элемента данных можно открыть, щелкнув по имени элемента данных или значку с тремя точками, в зависимости от раздела веб-интерфейса, например:

Monitoring > *Latest data*



Data Collection > *Templates* > *Items*



Содержимое

Меню элемента данных содержит три раздела: *Просмотр*, *Конфигурация* и *Действия*.

Раздел *Просмотр* содержит следующие параметры:

- **Последние данные** - открывает раздел *Последние данные*, отфильтрованный по текущему узлу сети и элементу данных;
- **График** - открывает простой график текущего элемента данных;
- **Значения** - открывает список всех значений, полученных для текущего элемента данных за последние 60 минут;
- **500 последних значений** - открывает список 500 последних значений текущего элемента данных.

Раздел *Конфигурация* (доступен только пользователям с типом *Администратор* и *Суперадминистратор*) содержит следующие параметры:

- **Элемент данных** - открывает форму настройки элемента данных для текущего элемента данных;
- **Шаблон** - открывает форму настройки шаблона, которому принадлежит элемент данных (доступно только при открытии меню элемента данных из раздела *Сбор данных* > *Шаблоны* > *Элементы данных*);
- **Узел сети** - открывает форму настройки узла сети, которому принадлежит элемент данных;
- **Триггеры** - при наведении указателя мыши открывает список триггеров элемента данных, если они есть; при нажатии на триггер открывается его форма настройки триггера;
- **Создать триггер** - открывает форму настройки триггера для создания триггера для этого элемента данных;
- **Создать зависимый элемент данных** - открывает форму настройки элемента данных для создания зависимого элемента данных, используя текущий элемент данных в качестве главного;
- **Создать зависимое правило обнаружения** - открывает форму настройки правила обнаружения для создания зависимого правила обнаружения, используя текущий элемент данных в качестве главного.

Раздел *Действия* содержит следующий параметр:

- **Выполнить сейчас** - немедленно выполняет проверку для получения нового значения элемента данных.

Поддерживаемые места

Меню элемента данных доступно при нажатии на имя элемента данных в различных разделах веб-интерфейса, например:

- *Мониторинг* > *Последние данные*
- *Сбор данных* > *Узлы сети* > *Элементы данных*

- *Сбор данных* > *Узлы сети* > *Правила обнаружения* > *Прототипы элементов данных*

Меню элемента данных доступно при нажатии на значение элемента данных в виджете панели мониторинга *Топ элементов данных*.

Разделы веб-интерфейса

Структура меню Меню веб-интерфейса Zabbix имеет следующую структуру:

- Панели
- Мониторинг
  - Проблемы
  - Узлы сети
  - Последние данные
  - Карты
  - Обнаружение
- APM
  - Трассировки
  - Метрики
  - Журналы
- Сервисы
  - Сервисы
  - SLA
  - Отчет по SLA
- Инвентаризация
  - Обзор
  - Узлы сети
- Отчеты
  - Сведения о системе
  - Запланированные отчеты
  - Отчет о доступности
  - 100 лучших триггеров
  - Журнал аудита
  - Журнал действий
  - Уведомления
- Сбор данных
  - Группы шаблонов
  - Группы узлов сети
  - Шаблоны
  - Узлы сети
  - Обслуживание
  - Корреляция событий
  - Обнаружение
- Оповещения
  - Действия
    - Действия триггеров
    - Действия сервисов
    - Действия обнаружения
    - Действия автоматической регистрации
    - Внутренние действия
  - Типы носителей
  - Скрипты
- Пользователи
  - Группы пользователей
  - Роли пользователей
  - Пользователи
  - Токены API
  - Аутентификация
  - Устройства
- Администрирование
  - Общее

        * Графический интерфейс
        * Автоматическая регистрация
        * Тайм-ауты
        * Изображения
        * Сопоставление значков
        * Регулярные выражения
        * Параметры отображения триггеров
        * Географические карты
        * Модули
        * Коннекторы
        * Прочее
    - Журнал аудита
    - Очистка данных
    - Группы прокси
    - Прокси
    - Макросы
    - Очередь
        * Обзор очереди
        * Обзор очереди по прокси
        * Подробности очереди

Панели мониторинга

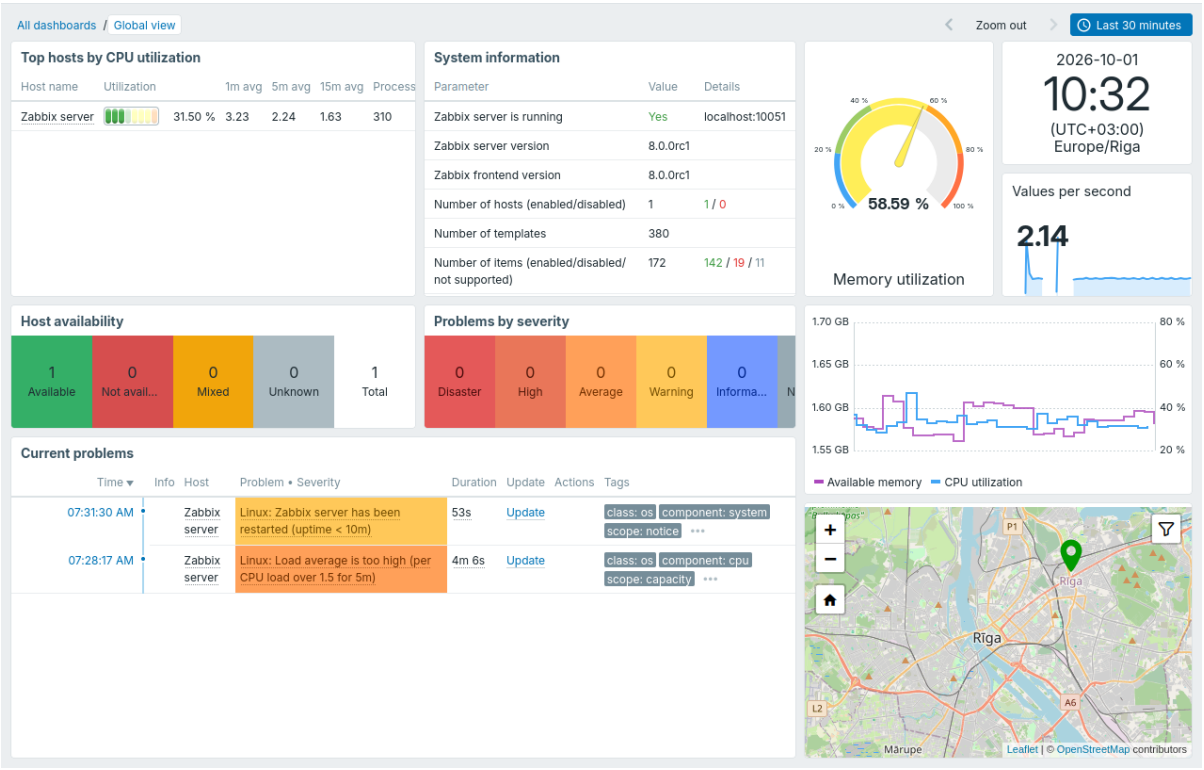
Панели мониторинга предоставляют обзор контролируемой среды. Вы можете создавать и настраивать их в соответствии со своими потребностями.

Основными элементами панелей являются виджеты: график, карта, список последних проблем и так далее. Каждый виджет предназначен для отображения определенного типа информации.

Виджеты можно организовать на нескольких страницах панели мониторинга. Такие страницы можно просматривать в режиме слайд-шоу. Также можно перейти на любую страницу, нажав ее вкладку.

Вы можете создать несколько панелей мониторинга, однако одновременно отображается только одна.

Zabbix поставляется с несколькими панелями мониторинга по умолчанию. Одна из них — панель мониторинга *Global view*, на которой отображаются работоспособность и состояние контролируемых узлов сети, включая сервер Zabbix:



Дополнительные панели мониторинга доступны в репозитории панелей мониторинга. Эти панели предназначены для работы с существующими шаблонами Zabbix. Их можно скачать и импортировать в вашу среду Zabbix. Подробное

описание каждой панели мониторинга, включая предварительные требования, доступно в файле README соответствующей панели.

Список панелей

Чтобы открыть список панелей, выполните одно из следующих действий:

- Нажмите *Панели* в меню.
- Нажмите *Все панели* во время просмотра панели, чтобы открыть список панелей.



Доступ к меню *Панели* зависит от роли пользователя и разрешения *Панель* в разделе *Доступ к элементам веб-интерфейса*.

Здесь можно: - Создавать панели. - Удалять панели. Чтобы удалить панель, установите флажок рядом с ней и нажмите кнопку *Удалить*. - Импортировать панели. - Экспортировать панели.

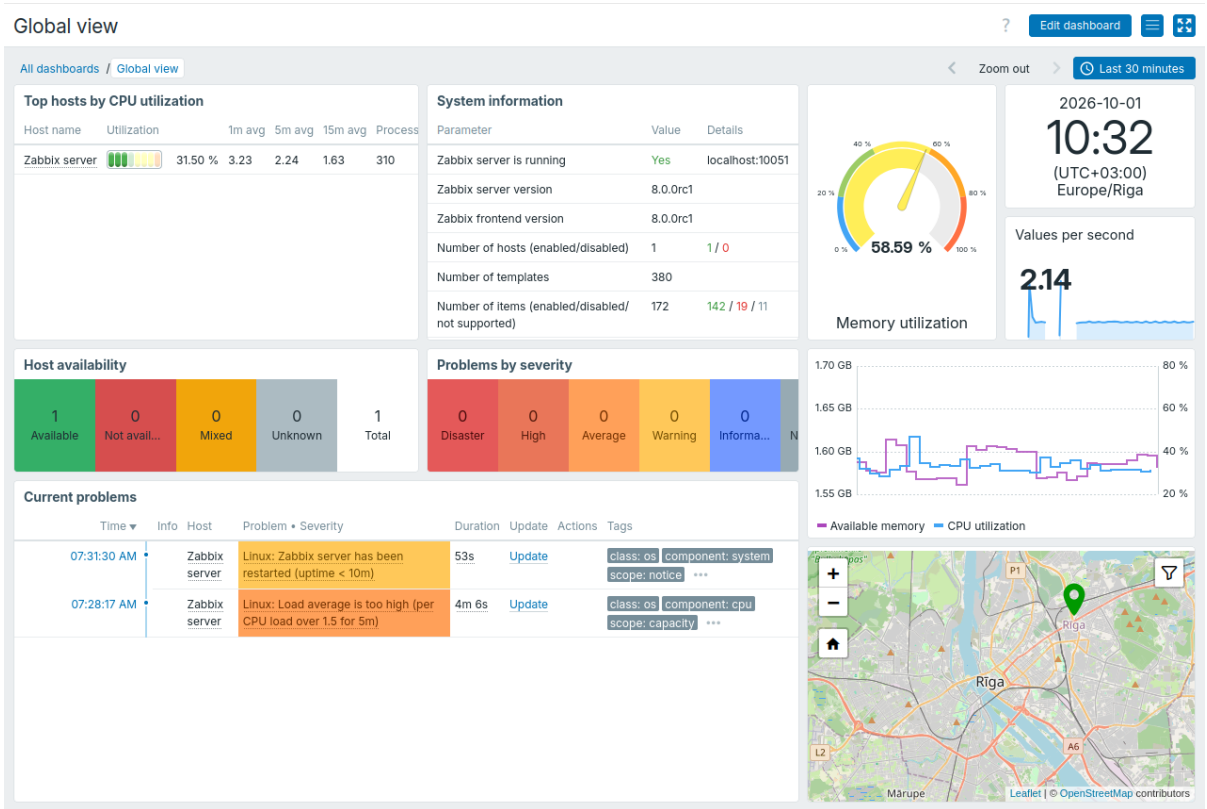
На панелях отображаются теги, указывающие их владельца и настройки общего доступа:

- *Мои* - панель принадлежит вам и видна только вам.
- *Мои + Общие* - панель принадлежит вам, но является общедоступной или доступна определенным пользователям или группам пользователей.
- Панели без тегов принадлежат другим пользователям, но являются общедоступными или доступны вам.

Список можно отфильтровать по имени панели и владельцу.

Просмотр панели

Нажмите имя панели в списке панелей, чтобы просмотреть панель:



Веб-интерфейс Zabbix запоминает последнюю открытую панель для следующего сеанса.

Note:
Данные, которые можно просматривать на панели, зависят от вашего доступа к узлам сети.

Для виджетов, отображающих графики, селектор периода времени над виджетами задает временной диапазон. Текущий временной диапазон отображается в виде метки. Нажмите метку, чтобы открыть или закрыть селектор.

Здесь можно управлять панелями следующим образом:

- Редактировать панель.
- Выполнять следующие действия из меню *Действия* панели :
  - *Общий доступ* - редактировать настройки общего доступа к панели.
  - *Создать новую* - создать новую панель.
  - *Клонировать* - создать новую панель путем клонирования текущей. Перед сохранением можно отредактировать свойства панели.
  - *Удалить* - удалить панель.
  - *Экспортировать* - экспортировать панель.
  - *Создать новый отчет* - создать отчет по расписанию. Этот параметр отключен, если у вас нет разрешения на управление отчетами по расписанию.
  - *Просмотреть связанные отчеты* - просмотреть список существующих отчетов для текущей панели. Этот параметр отключен, если связанные отчеты отсутствуют или у вас нет разрешения на управление отчетами по расписанию.
- Переключиться в режим киоска.
- Запустить или остановить слайд-шоу панели.

В режиме просмотра панели для виджетов доступны следующие параметры:

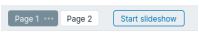
- Редактировать виджет.
- Выполнять следующие действия из меню *Действия* виджета :
  - *Копировать* - скопировать виджет.
  - *Скачать изображение* - скачать виджет в виде изображения PNG. Этот параметр доступен для виджетов, поддерживающих эту функцию (см. страницы отдельных виджетов).
  - *Интервал обновления* - выбрать частоту обновления содержимого виджета.

Для панелей с несколькими страницами в режиме просмотра панели доступны следующие действия в меню *Действия* страницы :

- *Копировать* - скопировать страницу.
- *Свойства* - настроить свойства страницы, включая имя и интервал слайд-шоу.

Слайд-шоу

Если флажок *Запускать слайд-шоу* выбран в свойствах панели мониторинга, слайд-шоу запускается автоматически, когда панель мониторинга содержит более одной страницы. В противном случае слайд-шоу можно запустить, нажав кнопку *Запустить слайд-шоу*.



Страницы сменяются в соответствии с интервалами, указанными в свойствах панели мониторинга и отдельных страниц. Если для отдельной страницы задано свойство *Интервал слайд-шоу*, применяется интервал, указанный для этой страницы. В противном случае применяется *Интервал слайд-шоу по умолчанию* панели мониторинга.

Чтобы остановить слайд-шоу, нажмите *Остановить слайд-шоу*.

В режиме киоска используйте следующие кнопки:

-  для запуска слайд-шоу;
-  для остановки слайд-шоу;
-  для перехода на предыдущую страницу;
-  для перехода на следующую страницу.

Общий доступ к панели мониторинга

Панели мониторинга могут быть общедоступными или частными.

Общедоступные панели мониторинга видны всем пользователям.
Частные панели мониторинга видны только их владельцам, а также пользователям или группам пользователей, которым предоставлен общий доступ.

Чтобы изменить настройки общего доступа к панели мониторинга, нажмите *Общий доступ* в меню *Действия* панели мониторинга во время просмотра панели мониторинга:

Параметр	Описание
Тип	Выберите тип панели мониторинга: **Частная** - панель мониторинга видна только выбранным группам пользователей и пользователям. **Общедоступная** - панель мониторинга видна всем пользователям.
Список общих доступов для групп пользователей	Выберите группы пользователей, которые могут получить доступ к панели мониторинга. Можно предоставить доступ только для чтения или для чтения и записи.
Список общих доступов для пользователей	Выберите пользователей, которые могут получить доступ к панели мониторинга. Можно предоставить доступ только для чтения или для чтения и записи.

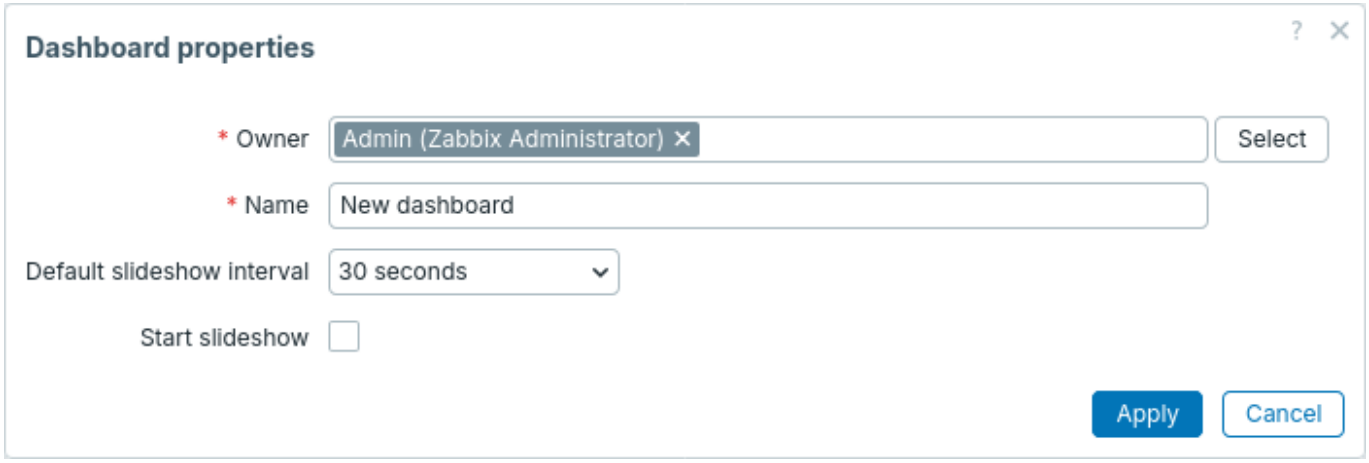
См. раздел Разрешения для панелей мониторинга, чтобы узнать, как параметры общего доступа ограничивают доступные пользователям действия.

Создание панели мониторинга

Панель мониторинга можно создать двумя способами:

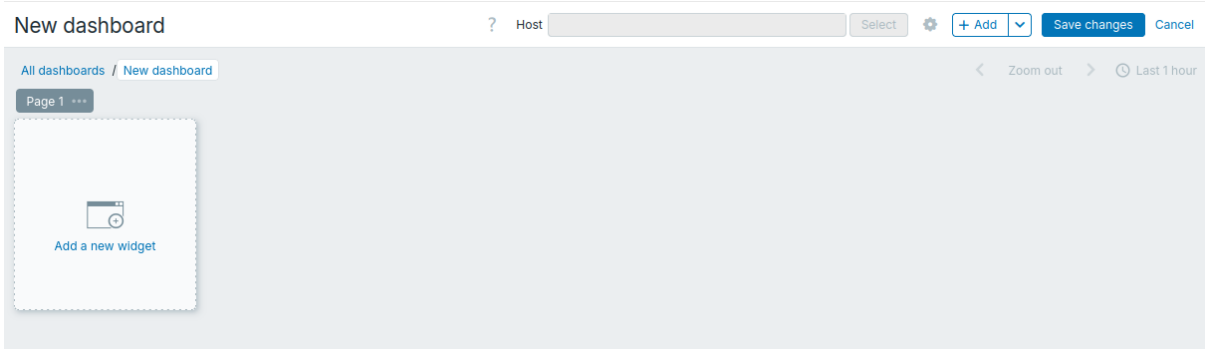
- В списке панелей мониторинга нажмите кнопку *Создать панель мониторинга*.
- В режиме просмотра панели мониторинга выберите *Создать новую* в меню *Действия* панели мониторинга.

Затем укажите свойства панели мониторинга:



Параметр	Описание
Владелец	Выберите пользователя, которому будет принадлежать панель мониторинга.
Имя	Введите имя панели мониторинга.
Интервал по умолчанию для слайд-шоу	Выберите, как долго будет отображаться каждая страница панели мониторинга, прежде чем слайд-шоу перейдет к следующей странице.
Запустить слайд-шоу	Установите этот флажок, чтобы автоматически запускать слайд-шоу, если панель мониторинга содержит более одной страницы.

Затем нажмите *Применить*. Новая панель мониторинга откроется в режиме редактирования:



Здесь можно добавить виджеты и страницы панели мониторинга.

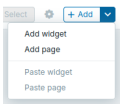
Обратите внимание, что минимальная ширина панели мониторинга составляет 1200 пикселей; в более узких окнах появляется горизонтальная полоса прокрутки. Более широкая панель мониторинга всегда заполняет окно браузера; ее виджеты не могут выходить за границы окна.

Нажмите кнопку *Сохранить изменения*, чтобы сохранить панель мониторинга, или *Отмена*, чтобы вернуться к предыдущему экрану.

Добавление виджетов

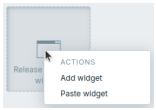
В режиме редактирования дашборда виджеты можно добавлять двумя способами:

- Нажмите кнопку *Добавить*. Также можно нажать стрелку раскрывающегося списка на кнопке *Добавить* и выбрать *Добавить виджет*.

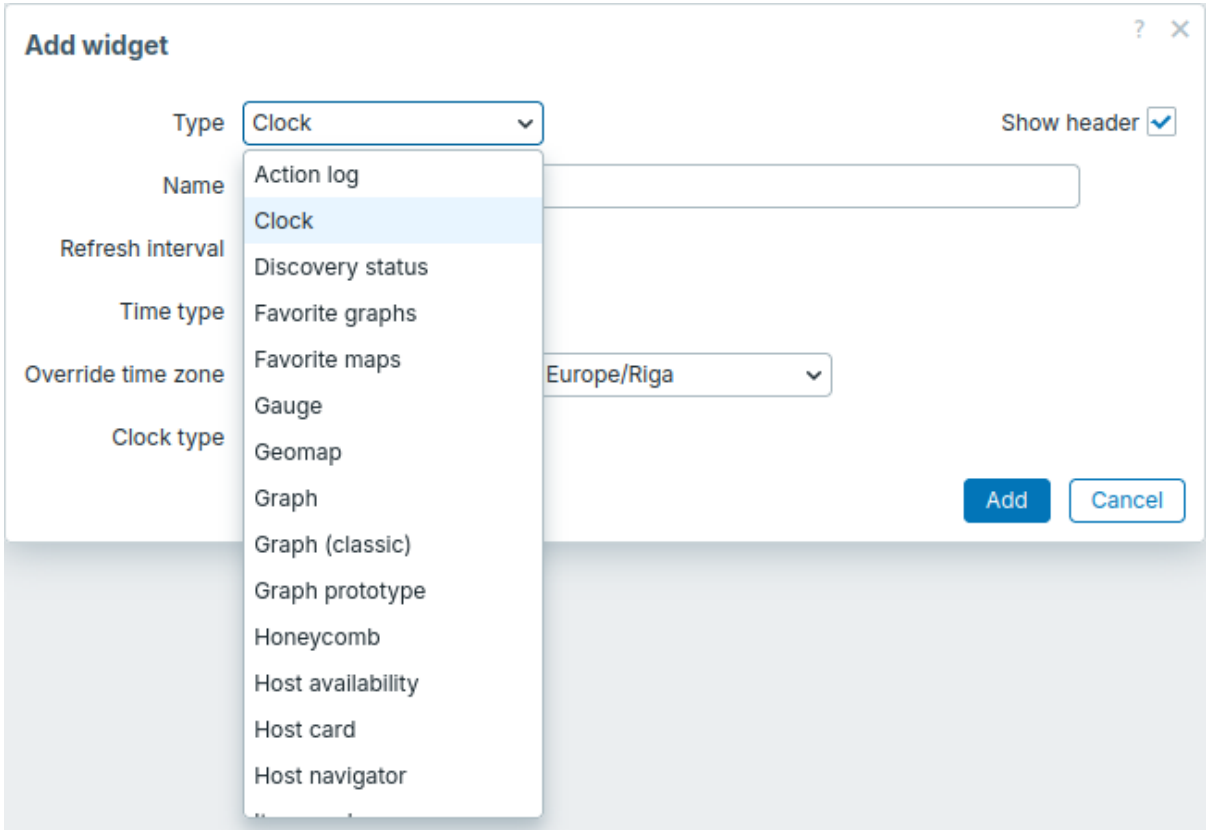
  Заполнитель виджета добавляется с размером по умолчанию в первое доступное свободное место на дашборде, после чего открывается форма настройки виджета.

- Наведите указатель мыши на свободную область дашборда. Затем нажмите на заполнитель виджета, чтобы открыть форму настройки виджета. Также можно нажать и перетащить заполнитель, чтобы изменить его размер, а затем отпустить кнопку мыши для открытия формы настройки виджета; если виджет скопирован в буфер обмена, сначала появится запрос на выбор действия *Добавить виджет* или *Вставить виджет*.

Затем выполните следующие действия:

1. В форме настройки виджета выберите тип виджета.
2. Настройте параметры виджета.
3. Нажмите *Добавить*, чтобы добавить виджет на дашборд.



Виджет добавляется с размером по умолчанию или изменяет размер в соответствии с доступным пространством. Изменить размер и положение виджетов можно в соответствии с описанием в разделе *Редактирование виджетов*.

Также можно копировать и вставлять виджеты, чтобы создавать новые виджеты с такой же конфигурацией — как в пределах одного дашборда, так и между дашбордами, открытыми в разных вкладках браузера.

Копирование и вставка виджетов

Чтобы скопировать и вставить виджет:

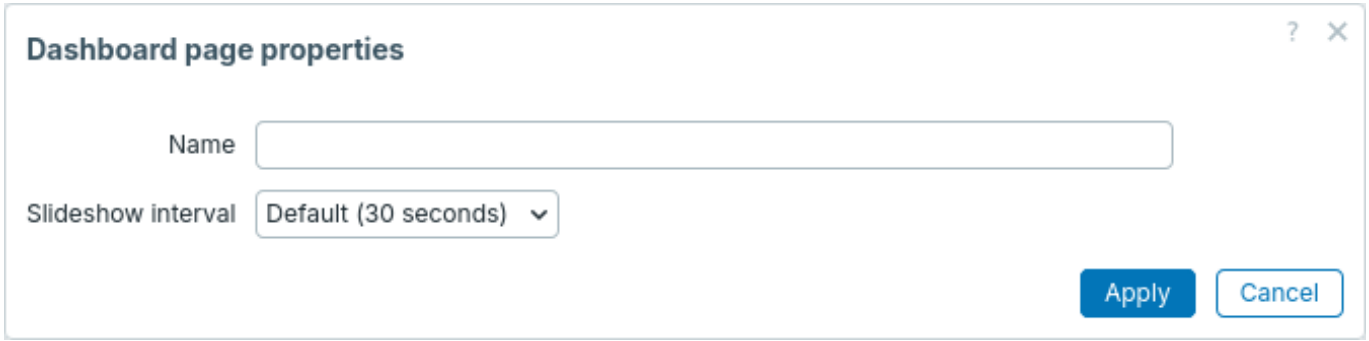
1. В меню *Действия* виджета ⋯ выберите *Копировать*.
2. В режиме редактирования панели мониторинга нажмите стрелку раскрывающегося списка на кнопке *Добавить* и выберите *Вставить виджет*. Также можно щелкнуть пустую область панели мониторинга и выбрать *Вставить виджет*.

Скопированный виджет также можно вставить поверх существующего, выбрав *Вставить* в его меню *Действия*.

Добавление страниц

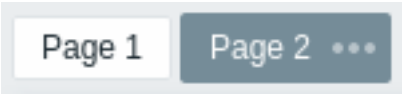
Чтобы добавить новую страницу на панель мониторинга:

1. В режиме редактирования панели мониторинга нажмите стрелку раскрывающегося списка на кнопке *Добавить* и выберите *Добавить страницу*.
2. Заполните свойства страницы:
   - *Имя* - введите имя страницы. Если оставить имя пустым, страница будет добавлена с именем *Страница N*, где «N» - порядковый номер страницы.
   - *Интервал слайд-шоу* - выберите, как долго страница будет отображаться, прежде чем слайд-шоу перейдет к следующей странице.



3. Нажмите *Применить*.

Для новой страницы будет добавлена новая вкладка (например, *Страница 2*).



Также можно копировать и вставлять страницы, чтобы создавать новые страницы с теми же свойствами, как в пределах одной панели мониторинга, так и между панелями мониторинга, открытыми в разных вкладках.

Копирование и вставка страниц

Чтобы скопировать и вставить страницу:

1. В меню *Действия* страницы выберите *Копировать*.
2. В режиме редактирования дашборда нажмите стрелку раскрывающегося списка на кнопке *Добавить* и выберите *Вставить страницу*.

Редактирование панели

Чтобы отредактировать панель, нажмите кнопку *Редактировать панель* в правом верхнем углу. Также можно нажать кнопку *Редактировать* для виджета. Панель откроется в режиме редактирования.

В этом режиме можно выполнять следующие действия с панелями:

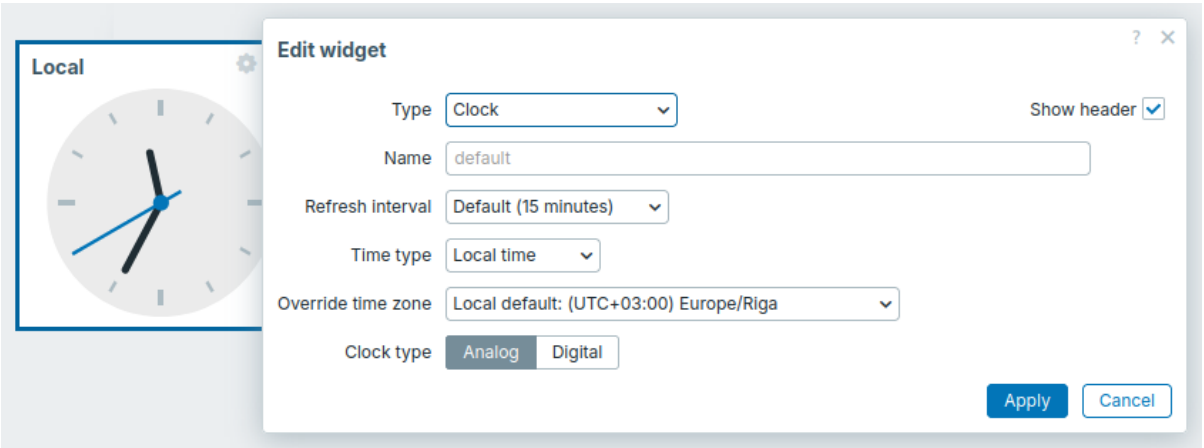
- Изменить свойства панели, нажав кнопку с изображением шестеренки.
- Редактировать виджеты панели.
- Получить доступ к следующим действиям с панелью, нажав стрелку раскрывающегося списка на кнопке *Добавить*:
  - *Добавить виджет* — добавить виджет на панель.
  - *Добавить страницу* — добавить страницу на панель.
  - *Вставить виджет* — вставить скопированный виджет. Этот параметр недоступен, если ни один виджет не был скопирован. Одновременно можно скопировать только один виджет.
  - *Вставить страницу* — вставить скопированную страницу. Этот параметр недоступен, если ни одна страница не была скопирована. Одновременно можно скопировать только одну страницу.

После редактирования панели нажмите кнопку *Сохранить изменения*, чтобы сохранить панель, или *Отмена*, чтобы вернуться к предыдущему экрану.

Редактирование виджетов

Виджет можно редактировать в режиме редактирования панели, нажав кнопку *Редактировать*. Кроме того, при просмотре панели можно навести указатель мыши на виджет и нажать кнопку *Редактировать*.

Во время редактирования виджет выделяется, а изменения отображаются в режиме реального времени.



Виджет и его форму настройки можно перемещать, перетаскивая их заголовки. Размер виджета можно изменять, перетаскивая маркеры изменения размера.



При редактировании виджета в его меню *Действия* доступны следующие действия:

- *Копировать* - скопировать виджет.

- *Вставить* - вставить скопированный виджет вместо этого виджета. Этот параметр недоступен, если ни один виджет не был скопирован.
- *Удалить* - удалить виджет.

Если обязательные параметры виджета не настроены, виджет переходит в состояние *Виджет настроен не полностью*.

После редактирования нажмите *Применить* в форме настройки виджета, а затем *Сохранить изменения* в правом верхнем углу панели, чтобы применить изменения.

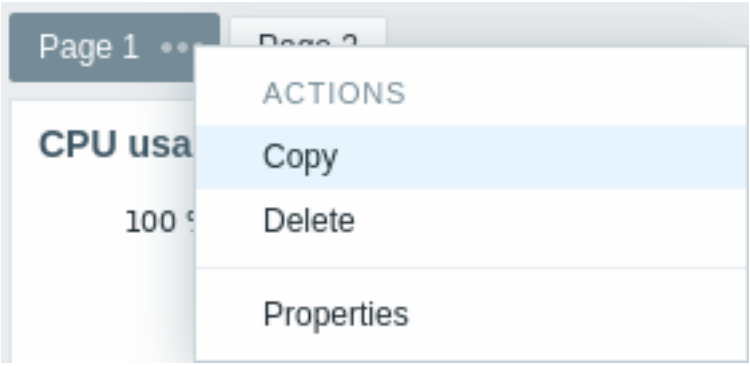
Редактирование страниц

Страницы можно редактировать в режиме редактирования дашборда.

Страницы можно менять местами, перетаскивая вкладки страниц. Изменение порядка не переименовывает страницы, за исключением страниц с автоматически назначенными именами (*Страница N*), номера которых изменяются.

При редактировании страницы в меню *Действия* доступны следующие действия:

- *Копировать* — скопировать страницу.
- *Удалить* — удалить страницу.
- *Свойства* — настроить свойства страницы.



Разрешения для панелей

Разрешения для панелей у обычных пользователей и пользователей типа *Admin* ограничены описанными ниже условиями, если для назначенной им роли пользователя включены доступ к панелям и параметр *Создание и редактирование панелей*:

- Они могут просматривать и клонировать общедоступную панель, даже если в настройках общего доступа к панели не добавлены пользователи или группы пользователей.
- Они могут просматривать и клонировать частную панель, если у них есть как минимум права *Чтение*, указанные в настройках общего доступа.
- Они могут редактировать и удалять панель только при наличии прав *Чтение-запись*, указанных в настройках общего доступа.
- Они не могут изменить владельца панели.

Виджеты панели мониторинга

Обзор

Zabbix предоставляет различные виджеты панели мониторинга, которые визуализируют и обобщают данные мониторинга, обеспечивая наглядный обзор производительности и доступности системы.

Ниже приведён обзор всех виджетов, сгруппированных по их основному назначению.

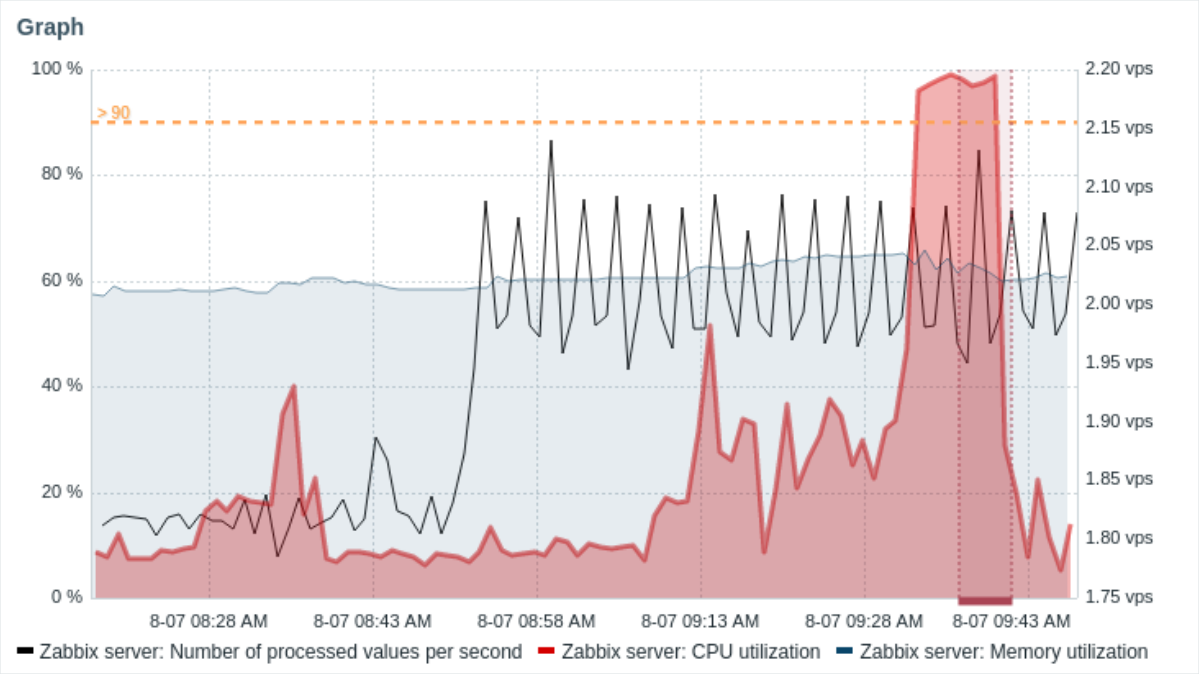
Подробные сведения о настройке виджетов см. в разделе Параметры виджетов панели мониторинга.

Визуализация данных

Эти виджеты представляют данные в различных визуальных форматах, помогая вам отслеживать данные и проблемы разными способами.

График

Виджет График отображает числовые данные элементов данных в виде векторного графика. Он помогает отслеживать метрики, выявлять проблемы и сравнивать значения во времени и между узлами сети.

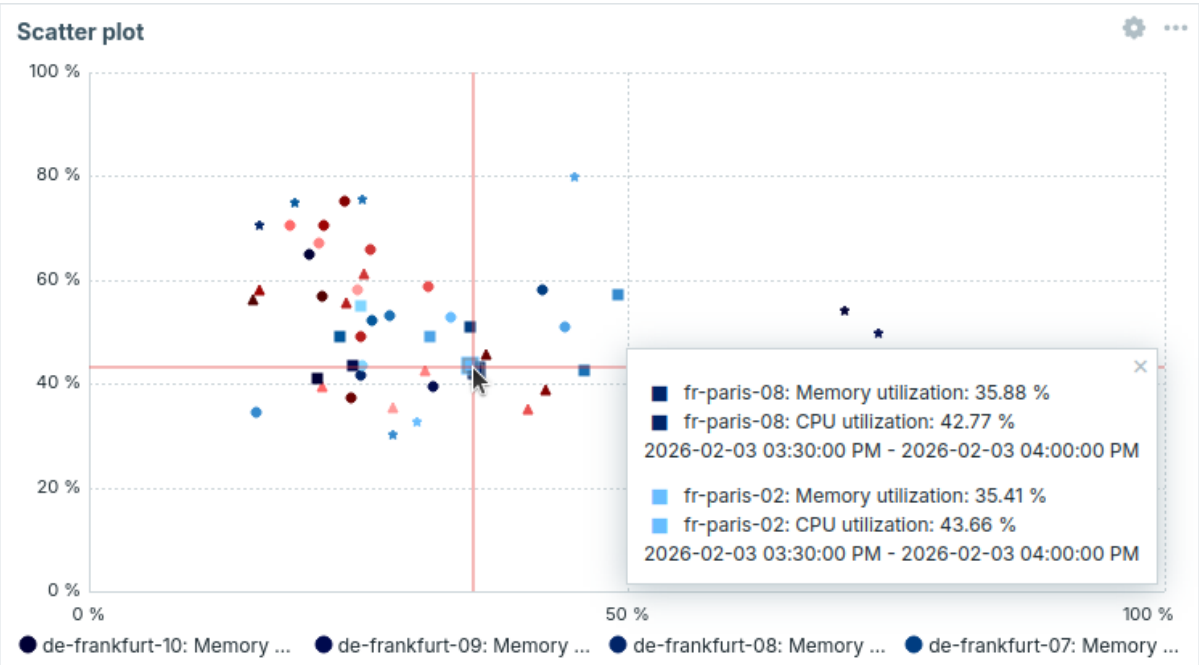


См. также:

- График (классический) — устаревший виджет графика, который отображает числовые данные элементов данных в виде графика на основе изображения.
- Прототип графика — отображает сетку автоматически созданных пользовательских графиков на основе прототипов графиков или прототипов элементов данных из правила обнаружения низкого уровня.
- Избранные графики — отображает алфавитный список ярлыков графиков, отмеченных текущим пользователем как избранные.

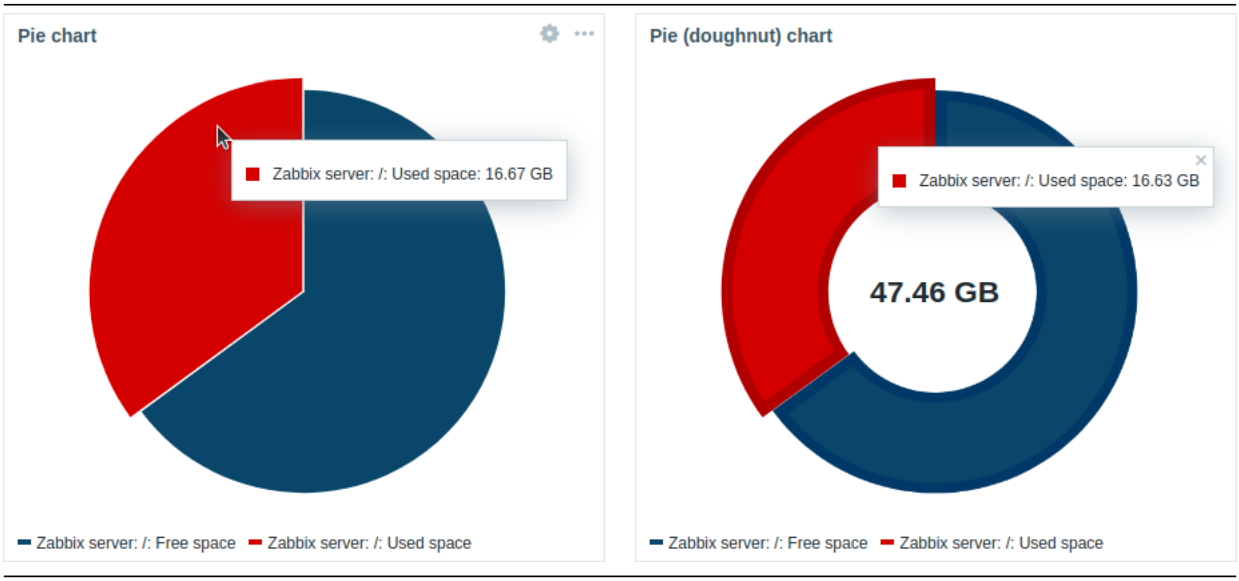
Диаграмма рассеяния

Виджет Диаграмма рассеяния отображает взаимосвязь между двумя метриками, нанося отдельные точки данных вдоль осей X и Y. Это помогает выявлять закономерности, кластеры, корреляции и выбросы в наборе данных.



Круговая диаграмма

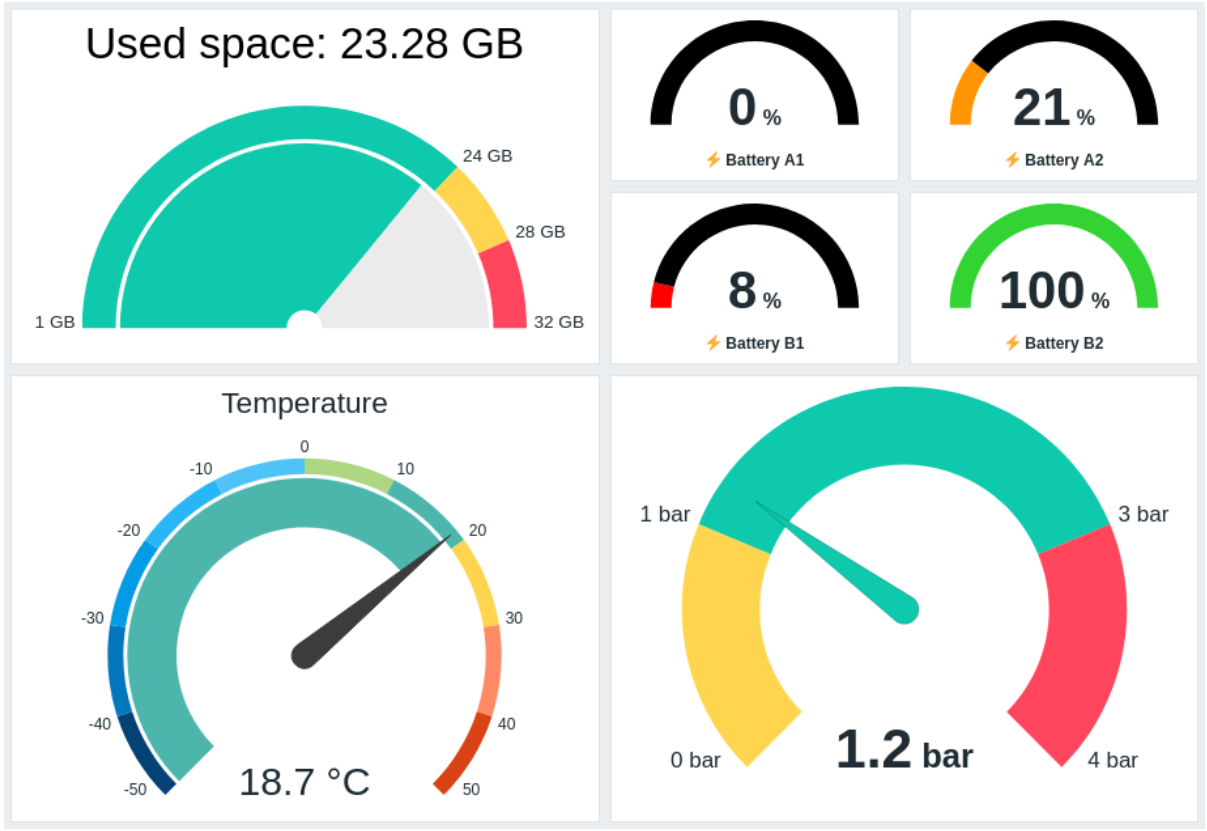
Виджет Круговая диаграмма отображает числовые данные элементов данных в виде векторной круговой или кольцевой диаграммы. Этот виджет помогает наглядно представить, как элементы данных или узлы сети вносят вклад в общий набор данных.



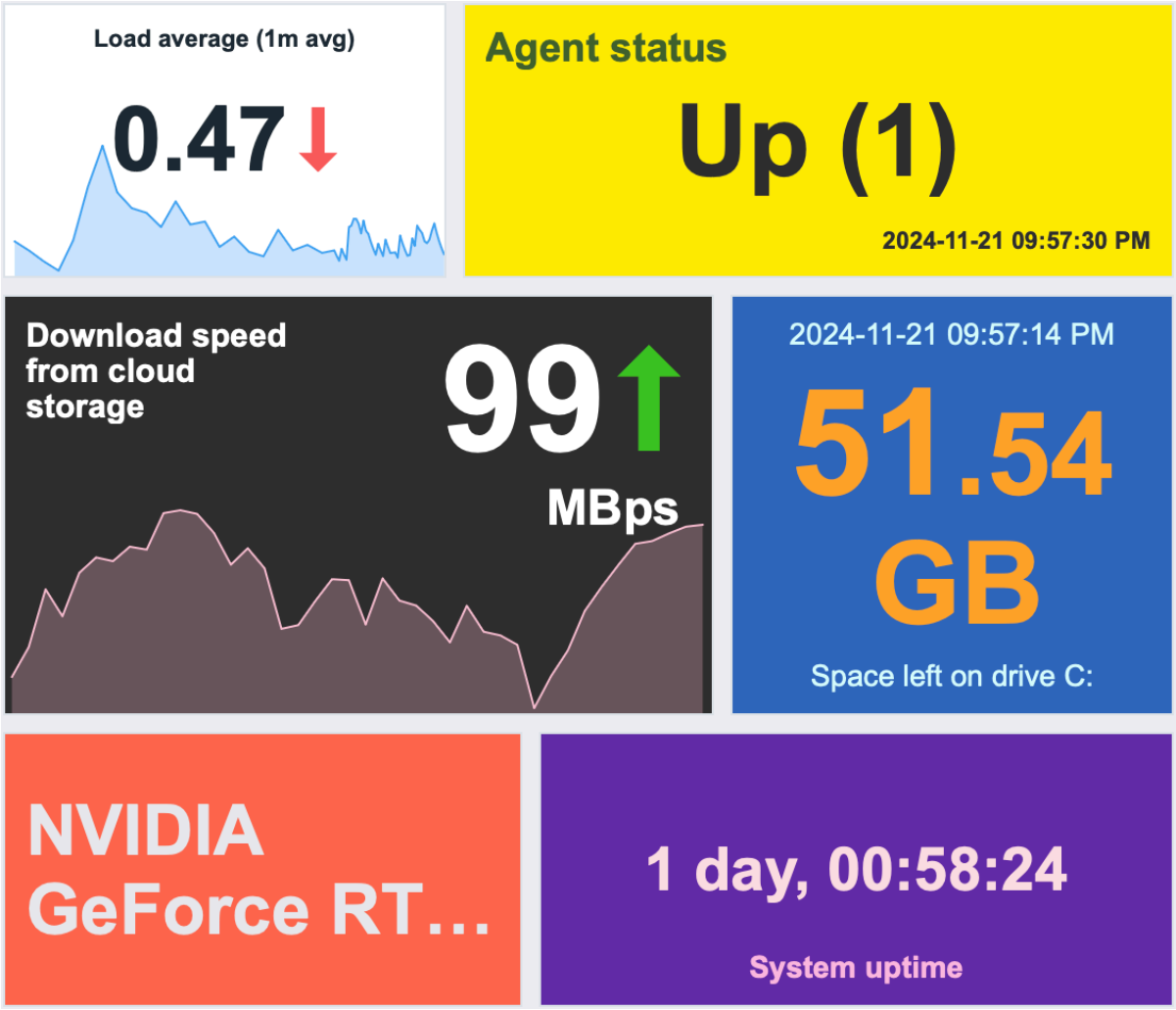
Gauge/Значение элемента данных

Виджеты Gauge/Значение элемента данных отображают значение одного элемента данных, что полезно для отслеживания важных метрик, визуализации пороговых значений и обнаружения резких изменений в данных.

Виджет Gauge:



Виджет «Значение элемента данных»:



История элемента данных

Виджет История элемента данных отображает последние данные для всех типов элементов данных (числовых, текстовых и т. д.) в формате таблицы. Он может показывать индикаторы выполнения и изображения (полезно для элементов данных браузера), а также может выделять значения (полезно для мониторинга файлов журналов).



Zabbix server

Timestamp	Name	Value
2024-05-30 01:54:24 PM	CPU utilization	100 %
2024-05-30 01:54:04 PM	Memory utilization	57.6091 %
2024-05-30 01:53:57 PM	Number of processed values per second	22.115
2024-05-30 01:53:24 PM	CPU utilization	100 %

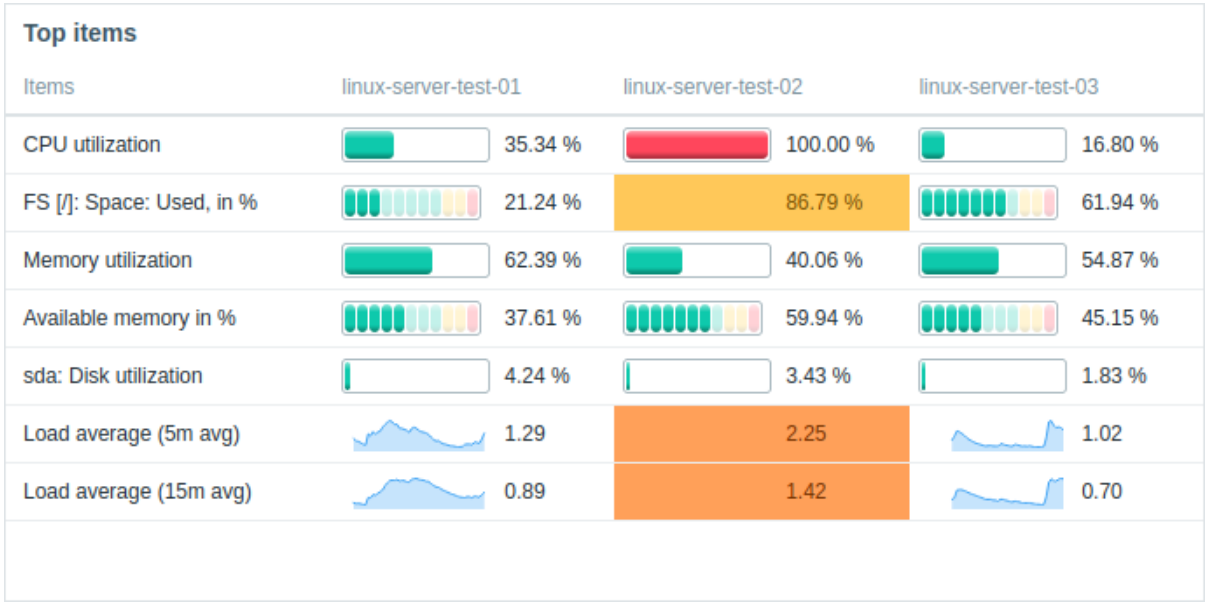
zabbix_agentd.log

```
7438:20240530:135401.322 zbx_setproctitle() title:'listener #1 [waiting for connection]'
8211:20240530:135401.321 zbx_popen(): executing script
7446:20240530:135401.320 zbx_setproctitle() title:'listener #9 [waiting for connection]'
7446:20240530:135401.320 Sending back [{"version":"7.0.0rc3","variant":1,"data":[{"error":"Accessible only as active check."}]}]
7446:20240530:135401.320 Requested [{"request":"passive checks","data":[{"key":"log[/tmp/zabbix_server.log,,,,skip]","timeout":4}]}]
```

Лучшие элементы данных

Виджет Top items отображает наибольшие/наименьшие значения для выбранных элементов данных, предоставляя

быстрый обзор их производительности.



См. также:

- Top hosts — отображает наибольшие/наименьшие значения для выбранных элементов данных с нескольких узлов сети.
- Top triggers — отображает триггеры с наибольшим количеством проблем.

Honeycomb

Виджет Honeycomb отображает узлы сети или элементы данных в виде шестиугольных ячеек в сетке. Такой макет позволяет легко выявлять проблемные узлы сети/элементы данных, замечать группы проблем, сравнивать группы с первого взгляда и многое другое.



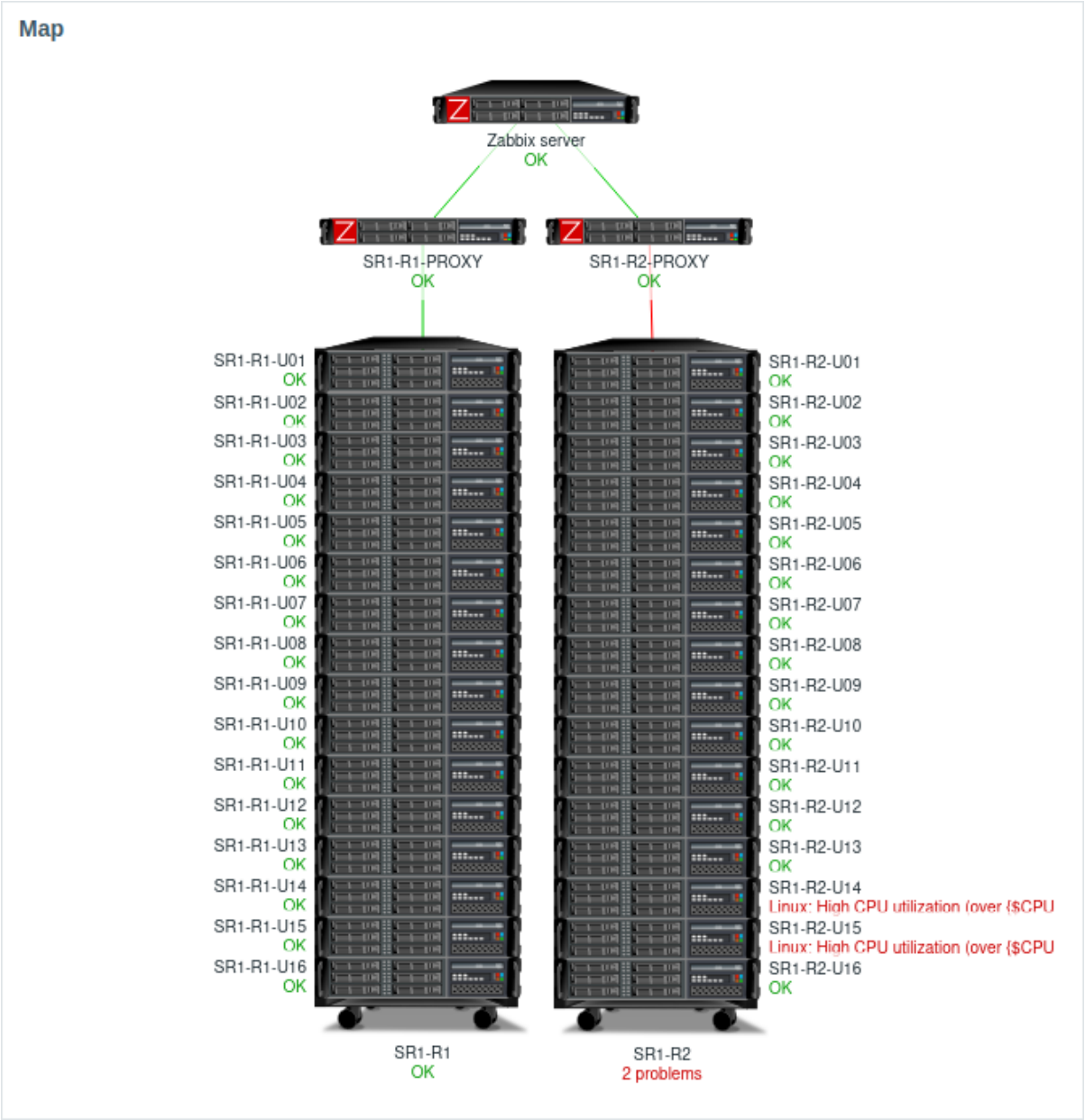


Карты

Виджеты карт визуализируют топологию сети или расположение узлов сети, помогая оценить общее состояние и состояние по конкретным местоположениям.

Карта

Виджет Карта может отображать карту сети или аналогичную визуализацию, предоставляя вам динамический обзор вашей сети.

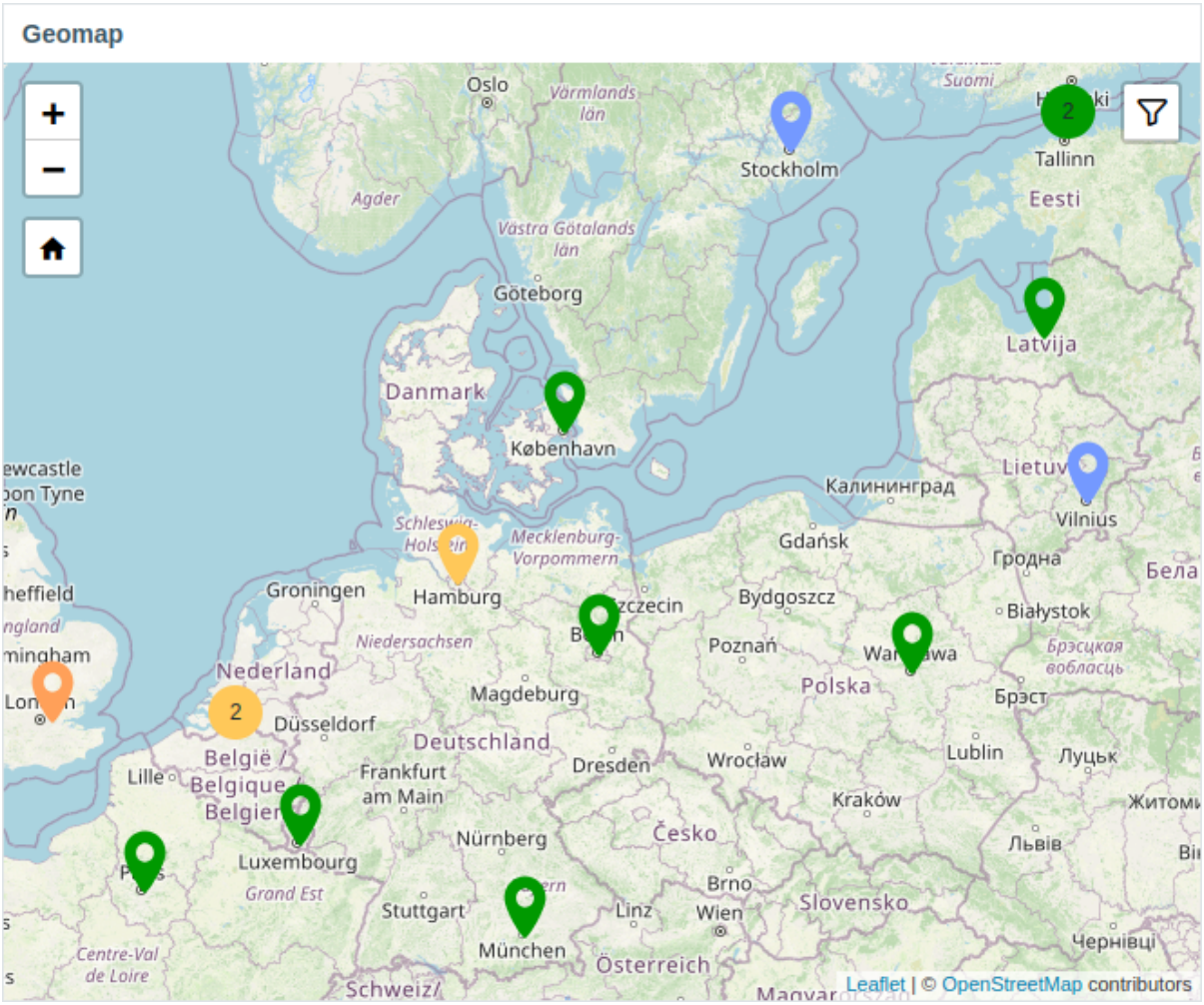


См. также:

- Дерево навигации карт — отображает иерархию существующих карт и количество проблем для каждой карты и группы карт.
- Избранные карты — отображает список ярлыков для карт, отмеченных текущим пользователем как избранные, отсортированный в алфавитном порядке.

Геокарта

Виджет Geomap отображает узлы сети в виде маркеров на интерактивной карте на основе Leaflet, при этом цвет маркеров указывает на наиболее серьёзную проблему каждого узла сети.

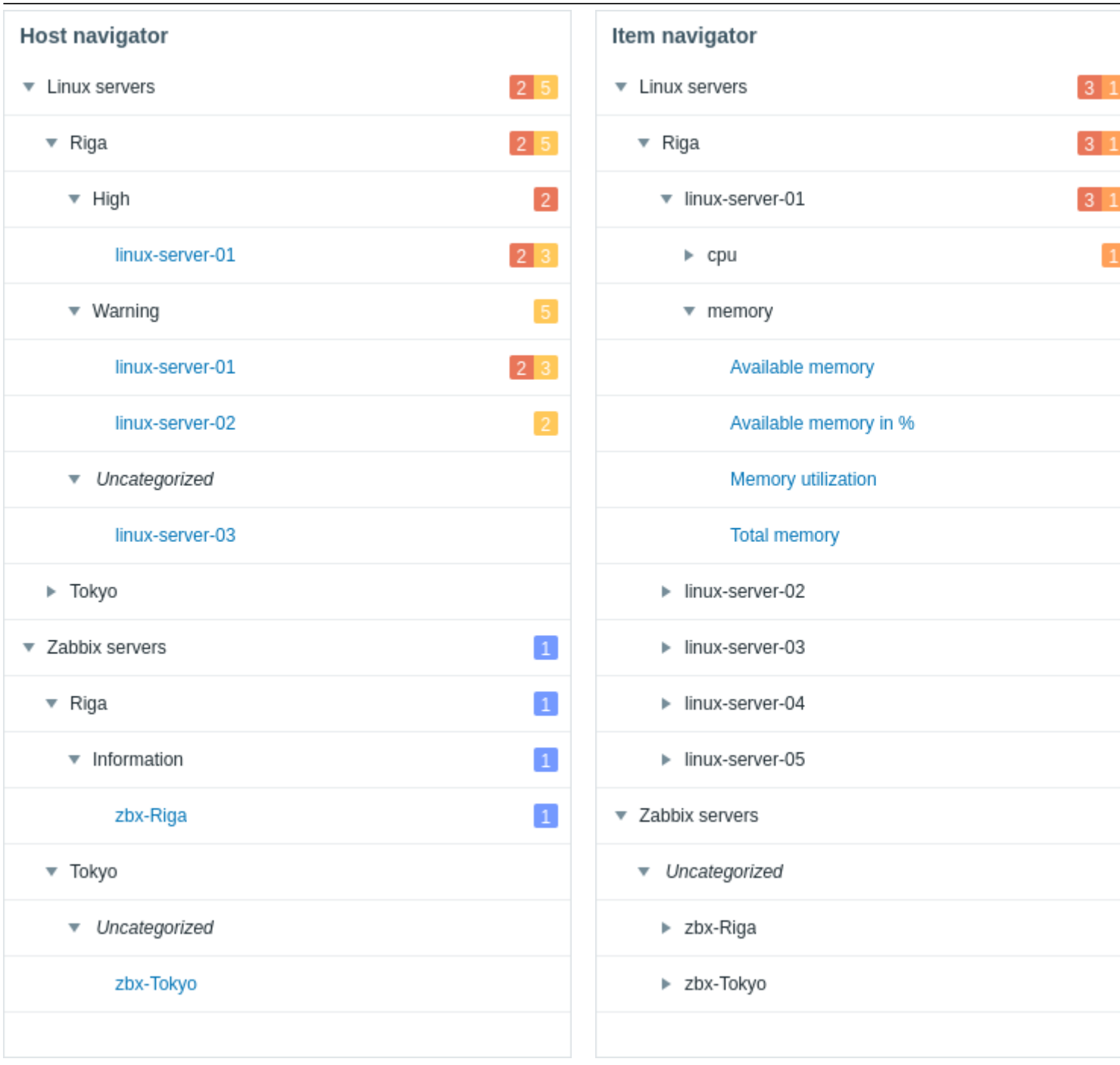


Сведения об узле сети/элементе данных

Эти виджеты предоставляют списки и подробные представления узлов сети или элементов данных.

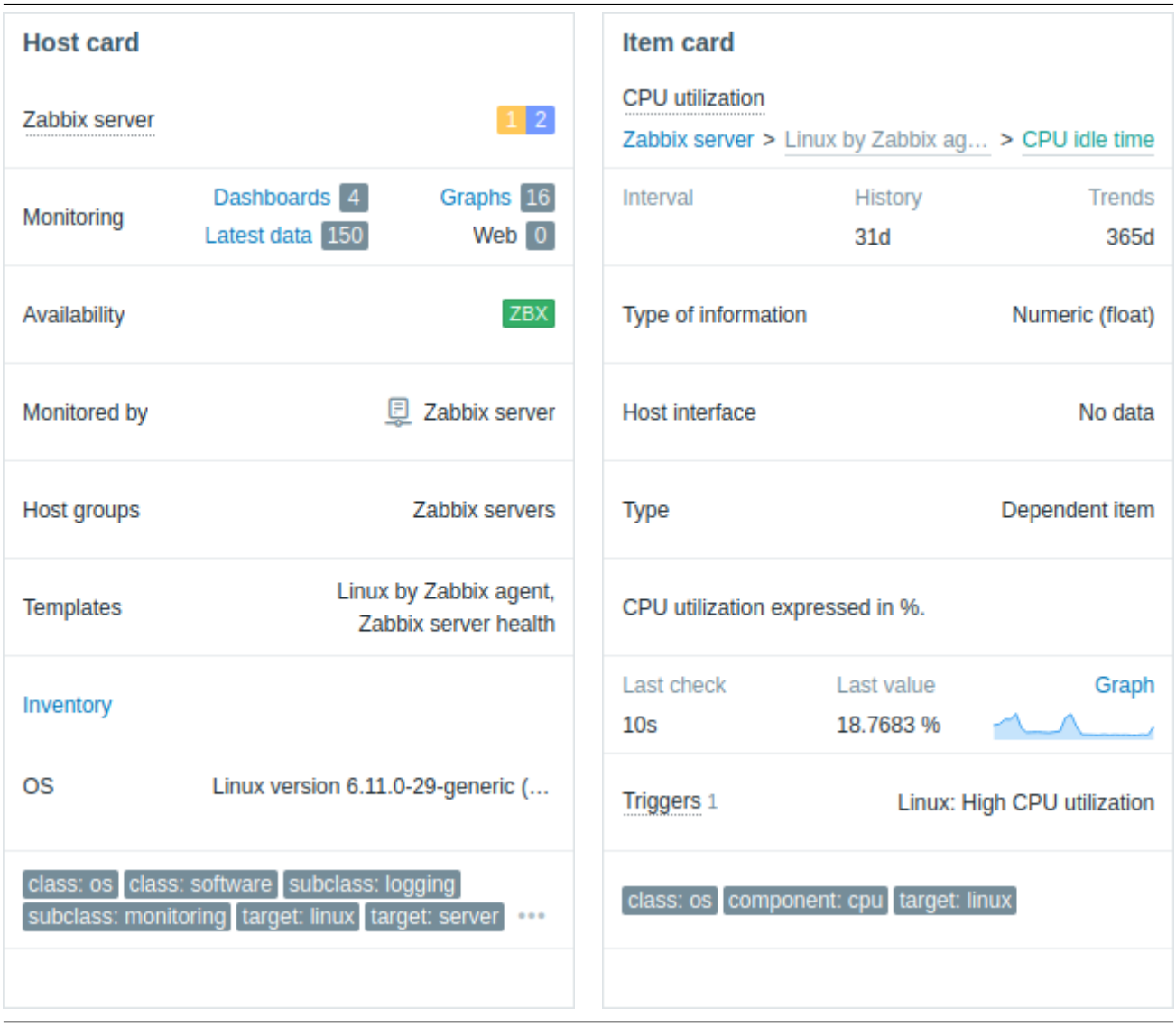
Навигатор узлов сети/элементов данных

Виджеты Навигатор узлов сети/Навигатор элементов данных отображают список узлов сети/элементов данных на основе различных параметров фильтрации и группировки. Эти виджеты наиболее полезны для управления тем, что отображают другие виджеты, на основе выбранного узла сети/элемента данных.



Карточка узла сети/элемента данных

Виджеты Карточка узла сети/Карточка элемента данных отображают сведения об одном узле сети/элементе данных, позволяя быстро оценить состояние и конфигурацию узла сети или элемента данных (доступность узла сети, проблемы, инвентарные данные, последние данные элемента данных, ошибки, описание и т. д.).



Состояние и отчёты

Эти виджеты отображают данные о состоянии (например, доступность, триггеры, проблемы), а также отчёты.

Доступность узлов сети

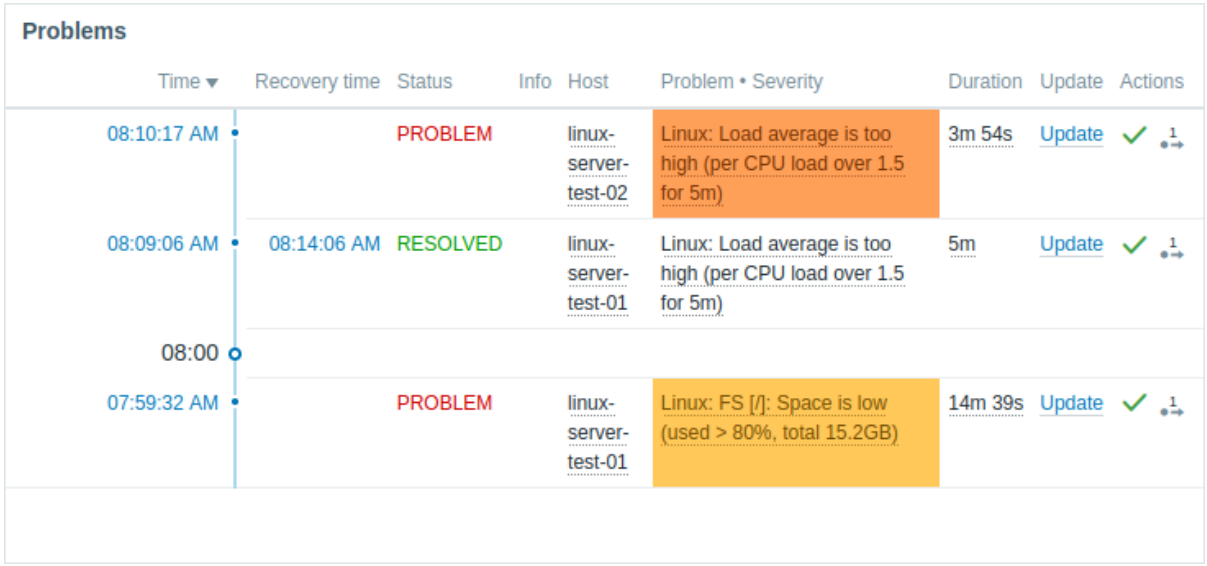
Виджет Доступность узлов сети отображает доступность узлов сети в выбранных группах узлов сети, позволяя отслеживать количество узлов сети, которые доступны или недоступны.

Host availability

	Total Hosts	Agent (active)	Agent (passive)	SNMP	JMX	IPMI
Available	10	3	8	2	0	0
Not available	2	0	3	0	0	0
Mixed	1	-	0	0	0	0
Unknown	4	0	4	0	0	0
Total	17	3	15	2	0	0

Проблемы

Виджет Проблемы отображает текущие проблемы, отфильтрованные по различным параметрам (группы узлов сети, узлы сети, имена проблем и т. д.), предоставляя наглядное представление о том, что требует внимания.



См. также:

- Проблемы по важности — отображает количество первопричинных проблем, сгруппированных по важности.
- Узлы сети с проблемами — отображает количество первопричинных проблем по каждой группе узлов сети и показывает наивысшую важность проблемы в каждой группе.

Обзор триггеров

Виджет Обзор триггеров отображает текущие состояния триггеров узлов сети в виде таблицы из цветных блоков. Он предоставляет быстрый наглядный обзор состояния и активности триггеров на различных узлах сети.



Журнал действий

Виджет Журнал действий отображает сведения об операциях (уведомлениях, удалённых командах), выполненных в рамках действия. Подробный журнал помогает при аудите, устранении неполадок и мониторинге выполнения каждого действия.

Action log

Time ▾	Action	Media type	Recipient	Message	Status	Info
2025-08-15 07:58:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Resolved in 1m 59s: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem has been resolved at 10:58:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Problem duration: 1m 59s Host: Linux server Severity: Average Original problem ID: 45	Sent	
2025-08-15 07:56:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Problem: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem started at 10:56:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Host: Linux server Severity: Average Operational data: Load averages(1m 5m 15m): (6.256836 4.808105 3.398438), # of CPUs: 2 Original problem ID: 45	Sent	
2025-08-15 07:42:30 AM	Report problems to Zabbix administrators		Admin (Zabbix Administrator)		Failed	i

Отчет SLA

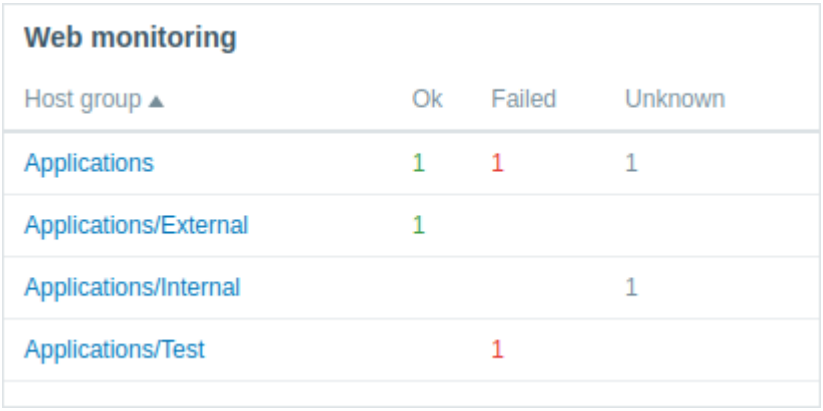
Виджет Отчет SLA отображает отчеты SLA, помогая вам контролировать производительность сервисов относительно заданных целевых показателей.

SLA report

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	99.6863	23h 55m 29s	4m 31s	-3m 5s	
2025-08-30	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-08-29	99.9%	99.7095	23h 55m 49s	4m 11s	-2m 45s	
2025-08-28	99.9%	99.9421	23h 59m 10s	50s	36s	
2025-08-27	99.9%	99.7188	23h 55m 57s	3m 15s	-2m 37s	
2025-08-26	99.9%	99.7755	23h 56m 46s	3m 14s	-1m 48s	
2025-08-25	99.9%	99.9094	22h 58m 45s	1h 1m 15s	8s	2025-08-25 01:00 AM Maintenance: 1h

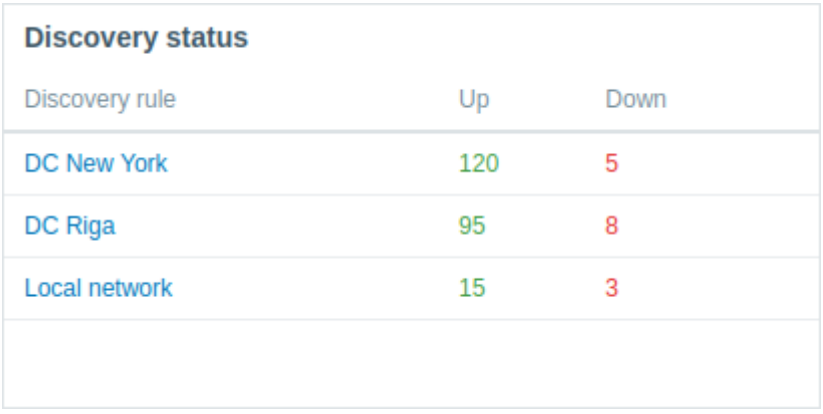
Веб-мониторинг

Виджет Веб-мониторинг отображает состояние активных веб-сценариев.



Статус обнаружения

Виджет Discovery status отображает состояние устройств, обнаруженных включенными правилами обнаружения сети. Он помогает отслеживать, корректно ли работает обнаружение сети, а также количество устройств, которые находятся в состоянии «доступен» или «недоступен».



Утилиты панели мониторинга и системная информация

Эти виджеты предоставляют дополнительную информацию и инструменты для расширения возможностей ваших панелей мониторинга.

Системная информация

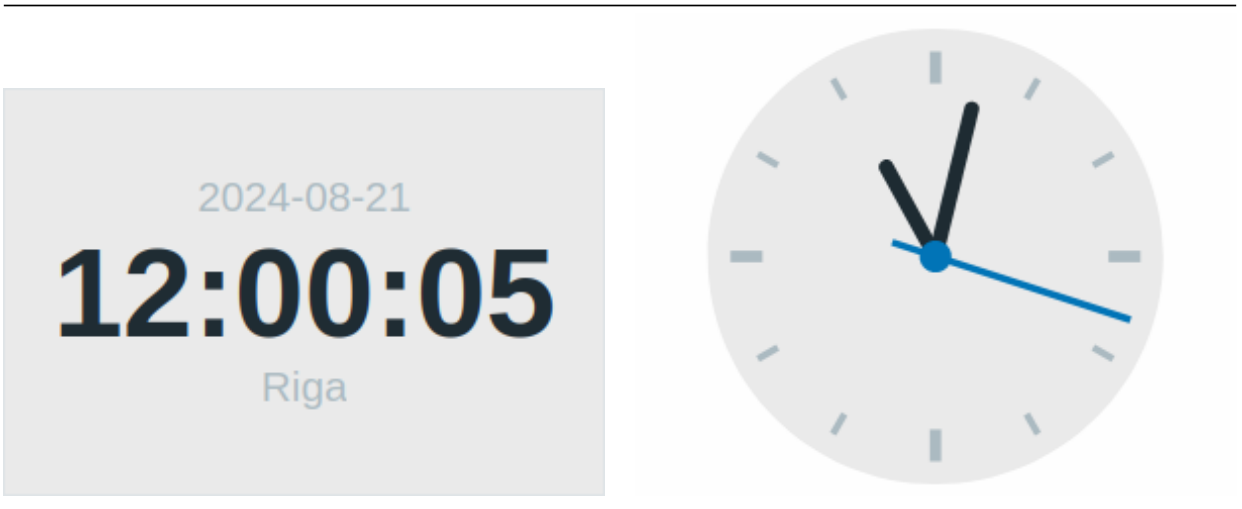
Виджет Системная информация отображает сводку по ключевым данным сервера Zabbix и системы или сведения об узлах высокой доступности.

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	8.0.0	Up to date
Zabbix frontend version	8.0.0	Up to date
Software update last checked	2025-08-30	
Latest release	8.0.0	Release notes
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	312	
Number of items (enabled/disabled/not supported)	176	165 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	89	89 / 0 [4 / 85]
Number of users (online)	2	1
Required server performance, new values per second	2.68	
Global scripts on Zabbix server	Disabled	
High availability cluster	Disabled	

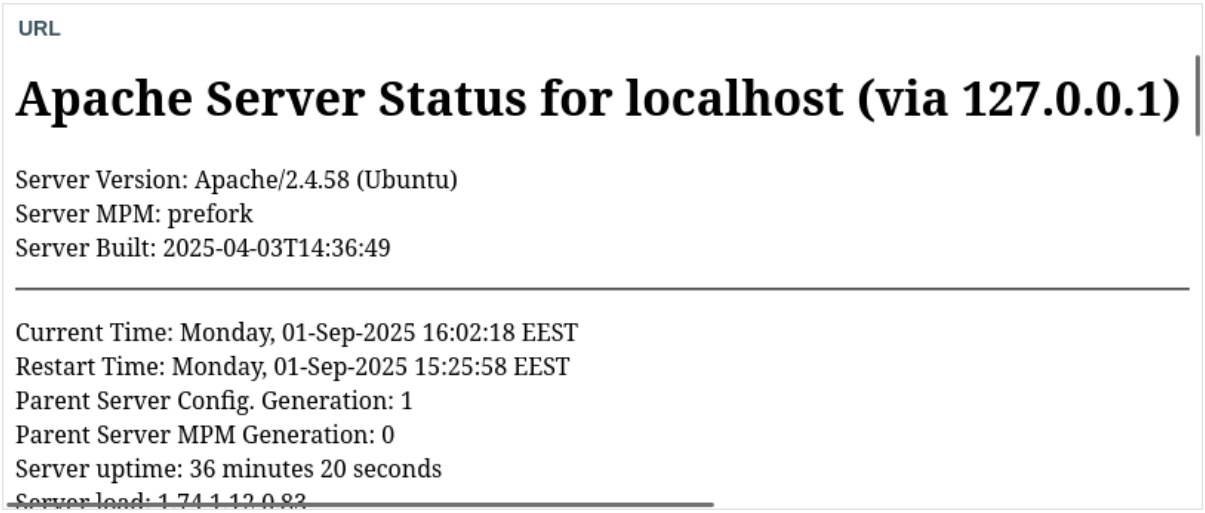
Часы

Виджет Clock отображает локальное время, время сервера или узла сети в аналоговом или цифровом формате.



URL

Виджет URL отображает содержимое, полученное по URL.



Параметры виджета панели мониторинга

Обзор

На этой странице описаны параметры, общие для всех виджетов панели мониторинга, а также динамические параметры, которые позволяют нескольким виджетам обмениваться данными конфигурации друг с другом или с панелью мониторинга.

Чтобы увидеть параметры, относящиеся к каждому конкретному виджету, воспользуйтесь боковой панелью для перехода на страницу соответствующего виджета.

Общие параметры

Следующие параметры являются общими для каждого виджета:

Имя	Введите имя виджета.
Интервал обновления	Настройте интервал обновления по умолчанию. Интервалы обновления виджетов по умолчанию варьируются от *Без обновления* до *15 минут* в зависимости от типа виджета. Например: - *Без обновления* для виджета *URL*; - *1 минута* для виджета *Журнал действий*; - *15 минут* для виджета *Часы*. Интервалы обновления можно задать в качестве значения по умолчанию для всех пользователей. Переключите панель мониторинга в режим редактирования, нажмите кнопку редактирования виджета и выберите нужный интервал обновления из раскрывающегося списка. Каждый пользователь также может задать собственный интервал обновления виджета. В режиме просмотра панели мониторинга нажмите кнопку с тремя точками на виджете и выберите нужный интервал обновления из раскрывающегося списка. Обратите внимание, что индивидуальный интервал обновления пользователя имеет приоритет над настройкой виджета и сохраняется даже после изменения настройки виджета.
Показывать заголовок	Установите флажок, чтобы постоянно отображать заголовок виджета. Если флажок не установлен, заголовок скрывается для экономии места и становится видимым только при наведении указателя мыши на виджет (как в режиме просмотра, так и в режиме редактирования). Заголовок также отображается полупрозрачно при перетаскивании виджета на новое место.

Динамические параметры

Несколько параметров виджета (например, *Узлы сети*, *Переопределить узел сети*, *Период времени*) поддерживают динамические источники данных. Вместо настройки статических значений вы можете связать эти параметры с другими виджетами или с самой панелью. После этого виджеты автоматически обновляются на основе выбранных значений или широковещательных сообщений из других частей панели.

Совместимость виджетов

Некоторые виджеты могут передавать данные конфигурации другим виджетам, некоторые могут принимать данные, а некоторые — и передавать, и принимать. Например:

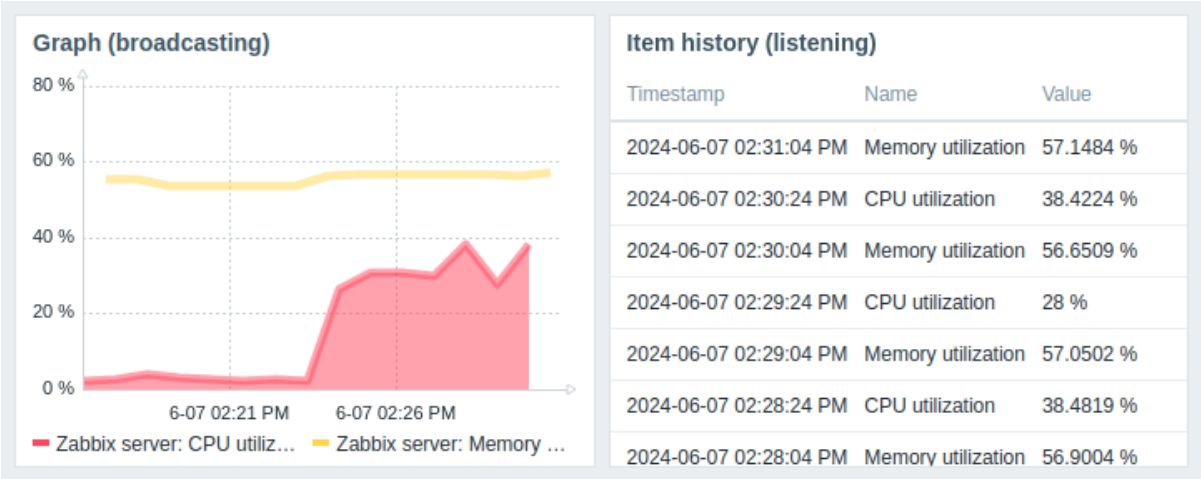
- Виджет *Action log* может получать данные о периоде времени только от виджетов *Graph*, *Graph (classic)* и *Graph prototype*.
- Виджет *Geomap* может передавать данные о узлах сети виджетам, которые их принимают (*Honeycomb*, *Top items* и т. д.), а также может принимать данные о группах узлов сети и узлах сети от виджетов, которые их передают (*Honeycomb*, *Problem hosts* и т. д.).
- Виджет *Clock* не может ни передавать, ни принимать какие-либо данные.

В следующей таблице описаны возможности каждого виджета по передаче и приему данных.

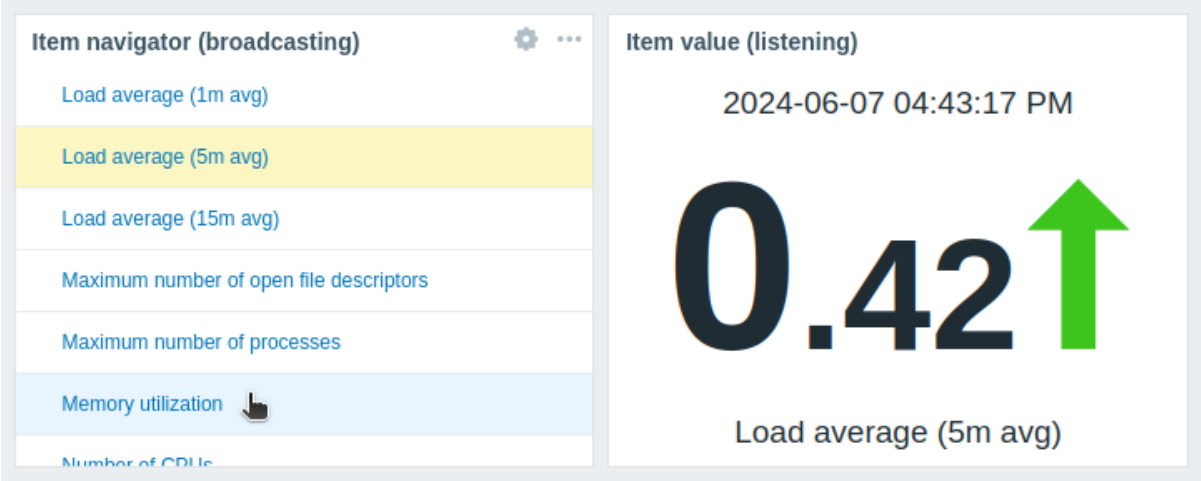
Widget	Broadcasts	Listens
Action log	-	Период времени
Clock	-	-
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	Узлы сети, Элементы данных
Geomap	Узлы сети	Группы узлов сети, Узлы сети
Graph	Период времени, Элементы данных	Узлы сети, Элементы данных, Период времени
Graph (classic)	Период времени	Узлы сети, Элементы данных, Графики, Период времени
Graph prototype	Период времени	Узлы сети, Период времени
Honeycomb	Узлы сети, Элементы данных	Группы узлов сети, Узлы сети
Host availability	-	Группы узлов сети
Host card	-	Узлы сети
Host navigator	Узлы сети	Группы узлов сети
Item card	-	Узлы сети, Элементы данных, Период времени
Item history	Элементы данных	Узлы сети, Период времени
Item navigator	Элементы данных	Группы узлов сети, Узлы сети
Item value	-	Узлы сети, Элементы данных, Период времени
Map	Группы узлов сети, Узлы сети	Карты
Map navigation tree	Карты	-
Pie chart	-	Элементы данных, Период времени
Problem hosts	Группы узлов сети	Группы узлов сети, Узлы сети
Problems	События	Группы узлов сети, Узлы сети
Problems by severity	Группы узлов сети	Группы узлов сети, Узлы сети
Scatter plot	Период времени, Элементы данных	Узлы сети, Элементы данных, Период времени
SLA report	-	-
System information	-	-
Top hosts	Узлы сети	Группы узлов сети, Узлы сети
Top items	-	Группы узлов сети, Узлы сети
Top triggers	-	Период времени
Trigger overview	-	Группы узлов сети, Узлы сети
URL	-	Узлы сети
Web monitoring	Группы узлов сети	Группы узлов сети, Узлы сети

Виджеты отличаются тем, как они **передают** данные другим виджетам.

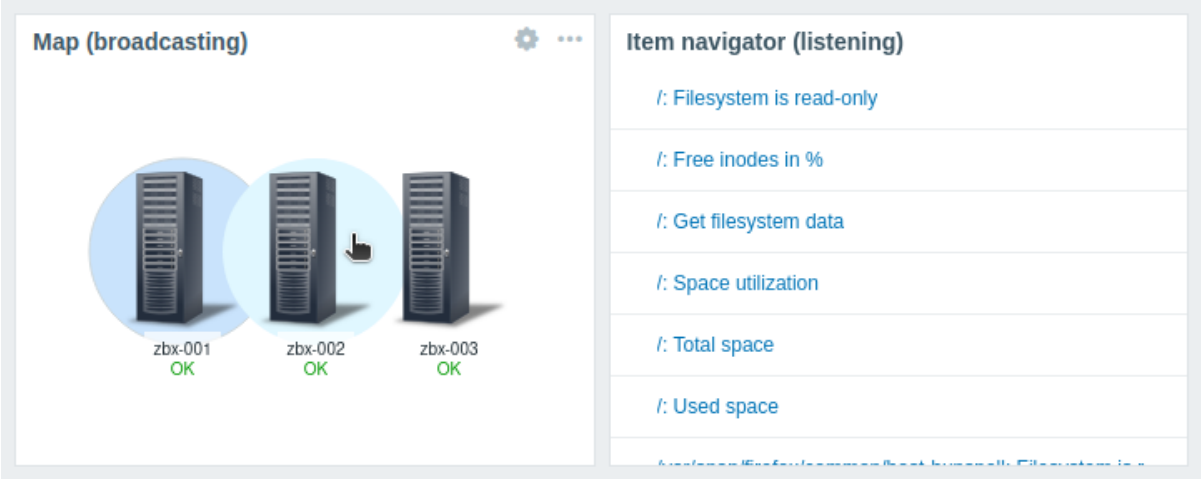
Виджеты, поддерживающие передачу данных, начинают делать это автоматически при создании. Например, виджет *Graph* сразу передает данные о временном периоде слушающим виджетам.



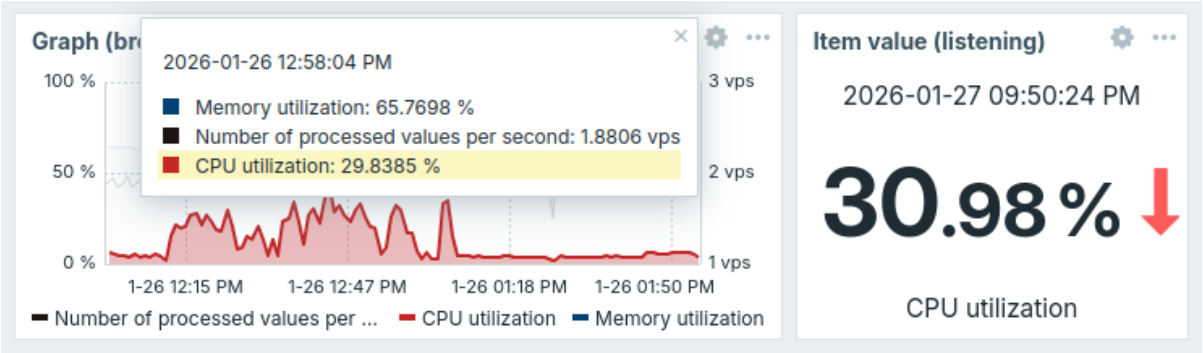
Виджеты, поддерживающие выбор сущности, автоматически передают данные для первой доступной сущности. Например, виджет *Item navigator* передает данные для первого элемента данных в своем списке элементов данных. Когда выбирается другой элемент данных, он передает данные для этого элемента данных. При наведении курсора элемент подсвечивается светло-синим; при выборе — желтым.



Виджет *Map* ведет себя аналогично, передавая данные для первого элемента, ближайшего к верхнему левому углу видимой области виджета (для виджета *Geomap* — ближайшего к центру). Когда выбирается другой элемент, он передает данные для этого элемента. При наведении курсора элемент подсвечивается светло-синим; при выборе — темно-синим.

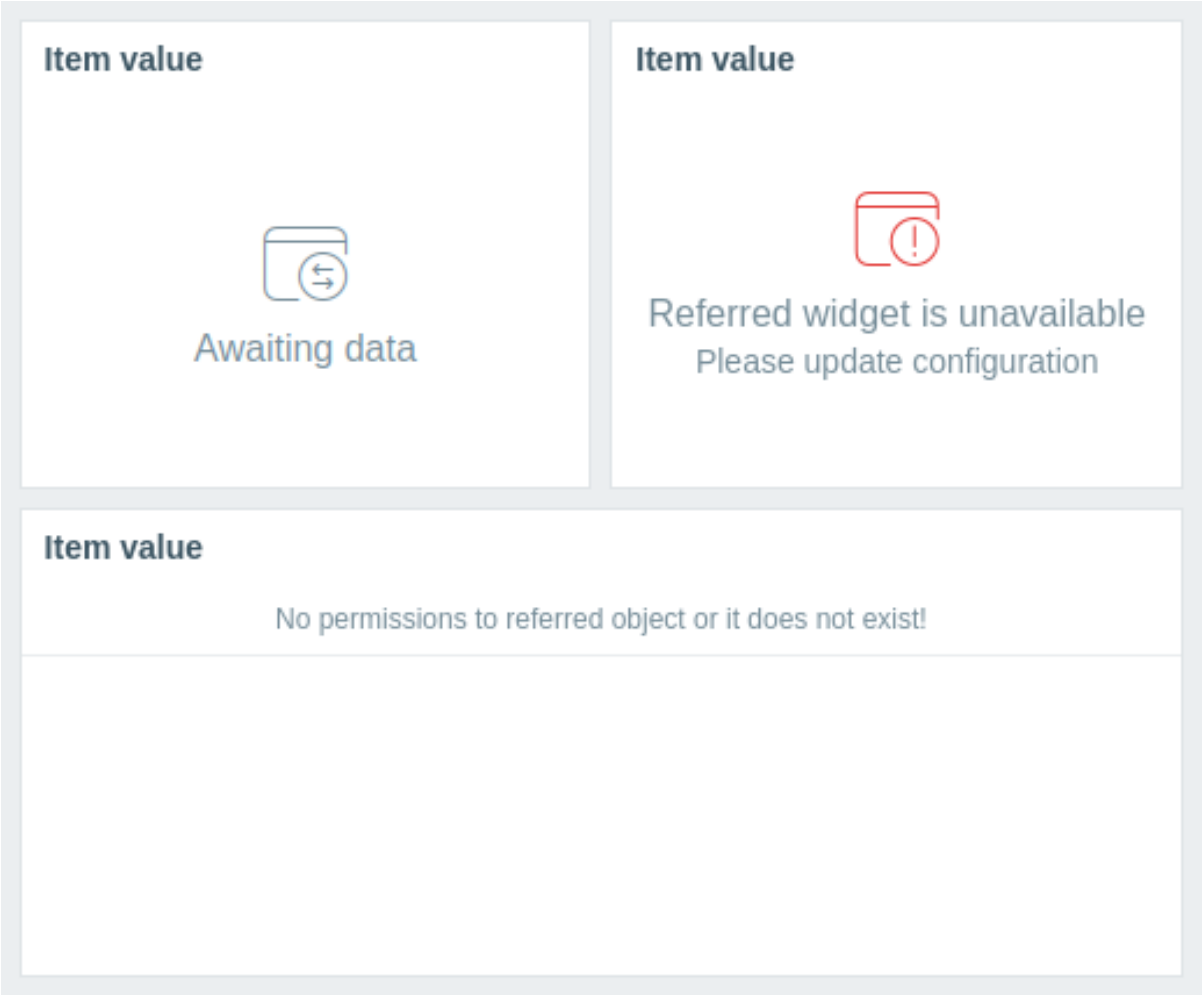


Такие виджеты, как *Graph* и *Scatter plot*, могут передавать данные элемента данных, когда элемент выбран в их всплывающих подсказках:



У виджетов также есть особое поведение при **получении** данных от других виджетов:

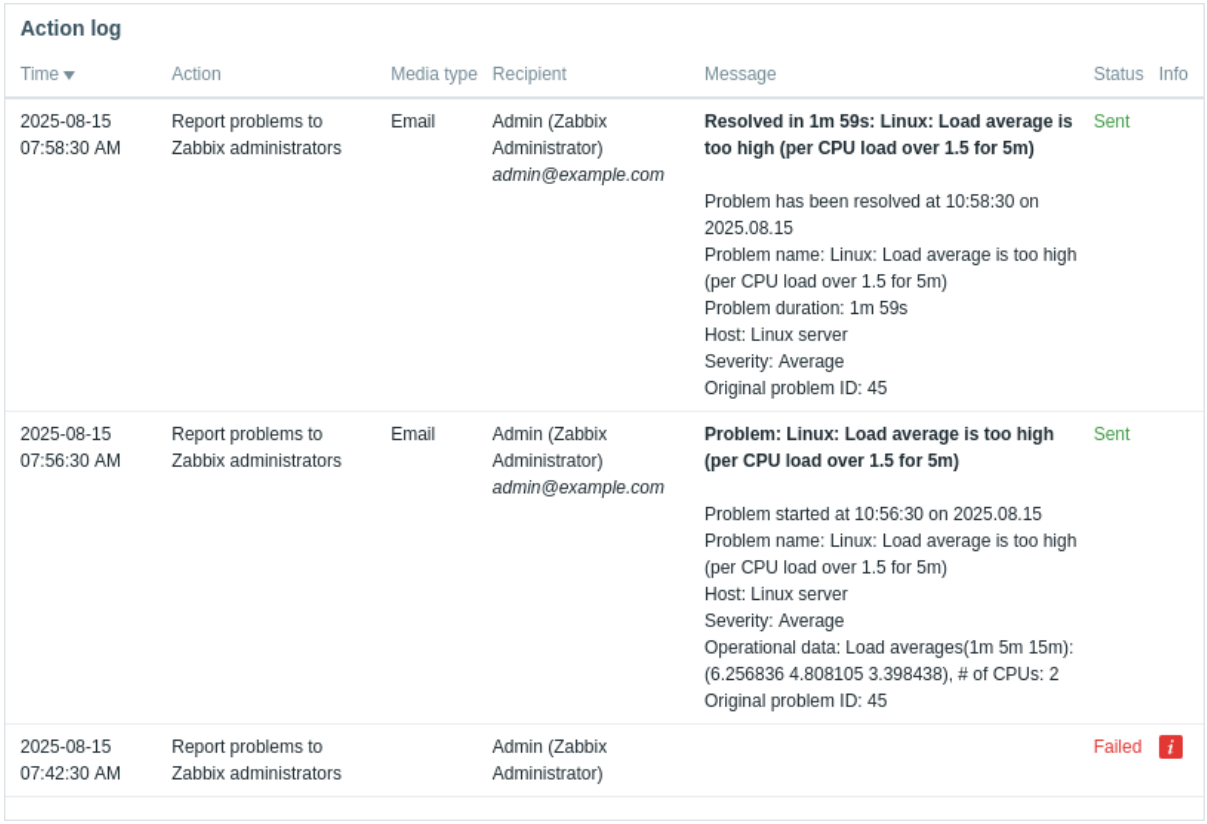
- Если виджет-источник данных не передает данные, принимающий виджет переходит в состояние *Awaiting data*.
- Если виджет-источник данных был удален, заменен несовместимым виджетом или перемещен на другую страницу дашборда, принимающий виджет переходит в состояние *Referred widget is unavailable*.
- Если указанный узел сети в источнике данных (виджете или дашборде) не содержит сущность, настроенную в принимающем виджете (элемент данных, график, карту и т. д.), или если у пользователя нет прав доступа к узлу сети, принимающий виджет отображает следующее сообщение: *"No permissions to referred object or it does not exist!"*



Журнал действий

Обзор

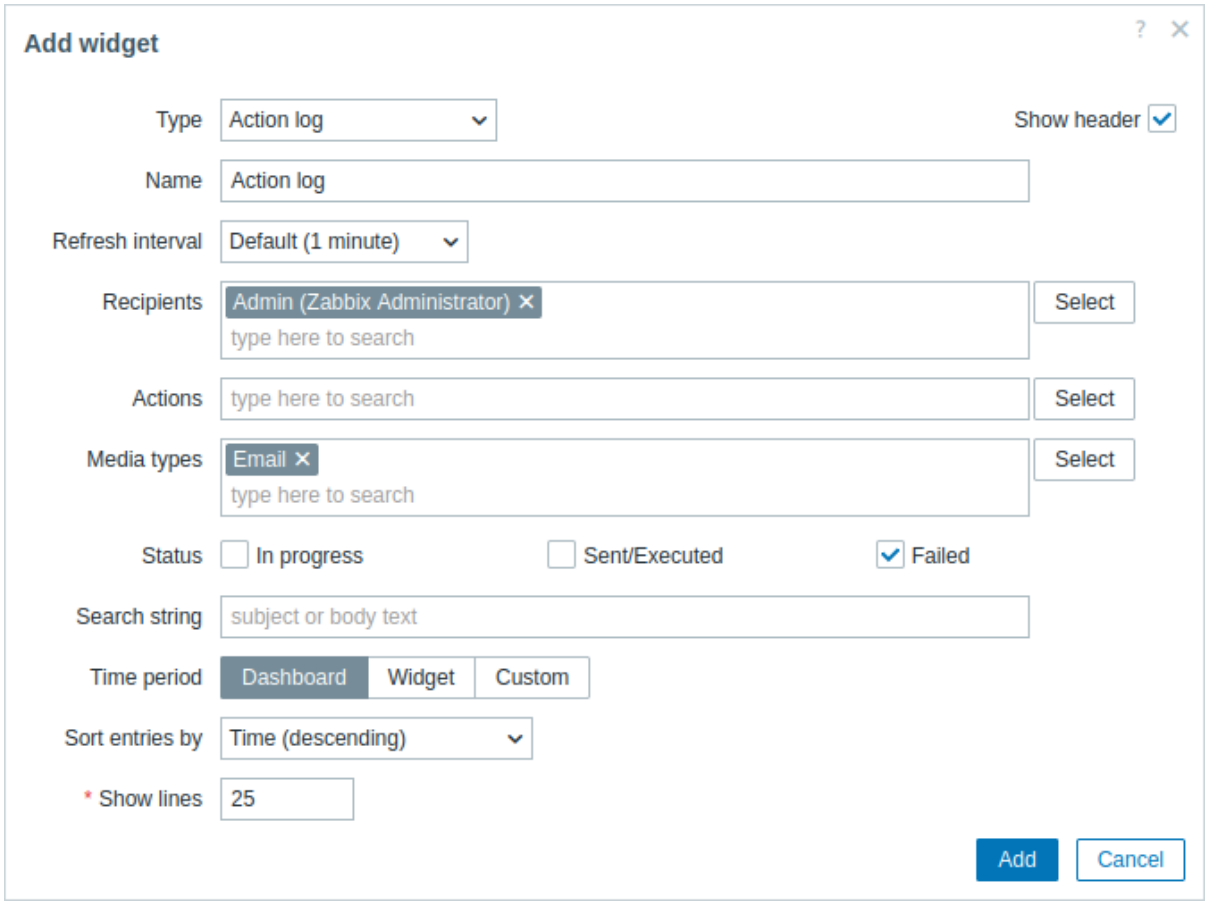
Виджет *Журнал действий* отображает сведения об операциях (уведомлениях, удаленных командах), выполненных в рамках действия. Подробный журнал помогает при аудите, устранении неполадок и мониторинге выполнения каждого действия.



Он отображает те же данные, что и *Отчеты* > *Журнал действий*, и может показывать до 1000 записей.

Конфигурация

Чтобы настроить, выберите *Action log* в качестве типа:



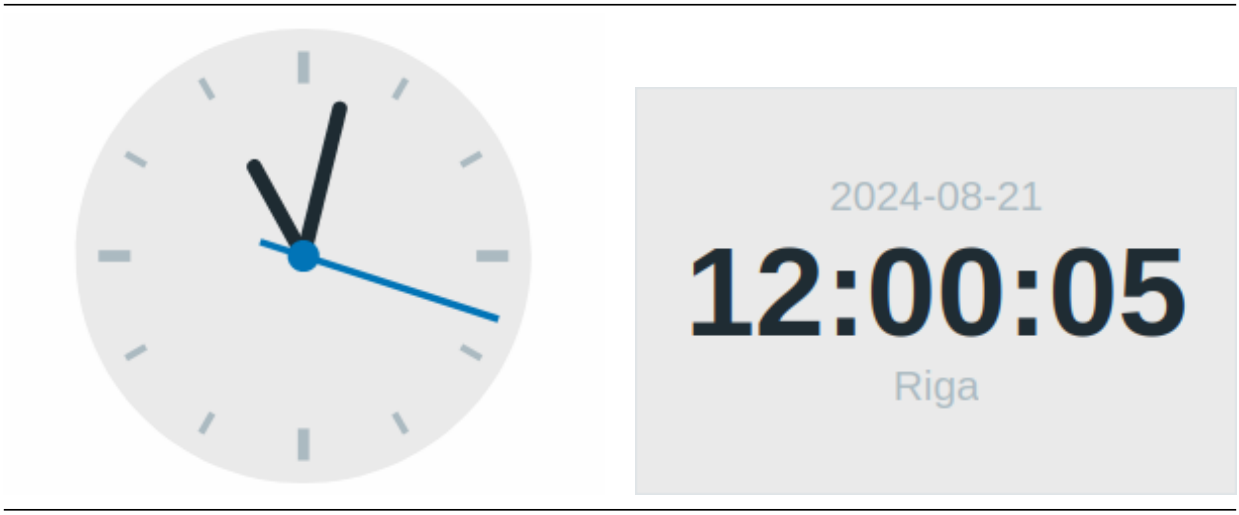
В дополнение к параметрам, общим для всех виджетов, можно задать следующие специальные параметры:

Recipients	Фильтрация записей по получателям. Это поле поддерживает автодополнение, поэтому при вводе имени получателя будет предлагаться выпадающий список подходящих получателей. Если получатели не выбраны, будут отображаться сведения о действиях для всех получателей.
Actions	Фильтрация записей по действиям. Это поле поддерживает автодополнение, поэтому при вводе имени действия будет предлагаться выпадающий список подходящих действий. Если действия не выбраны, будут отображаться сведения о действиях для всех действий.
Media types	Фильтрация записей по типам медиа. Это поле поддерживает автодополнение, поэтому при вводе имени типа медиа будет предлагаться выпадающий список подходящих типов медиа. Если типы медиа не выбраны, будут отображаться сведения о действиях для всех типов медиа.
Status	Установите флажок, чтобы фильтровать записи по соответствующему статусу: **In progress** - отображаются выполняющиеся действия; **Sent/Executed** - отображаются действия, для которых было отправлено уведомление или которые были выполнены; **Failed** - отображаются действия, завершившиеся с ошибкой.
Search string	Фильтрация записей по содержимому сообщения/удаленной команды. Если вы введете здесь строку, будут отображаться только те действия, в содержимом сообщения/удаленной команды которых присутствует введенная строка. Макросы не подставляются.
Time period	Фильтрация записей по временному периоду. Выберите источник данных для временного периода: **Dashboard** - использовать селектор временного периода панели; **Widget** - использовать совместимый виджет (задается в параметре *Widget*); **Custom** - использовать пользовательский временной период, заданный в параметрах *From* и *To*; если задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты по-прежнему могут использовать его как источник данных для временного периода.
Widget	Введите или выберите совместимый виджет в качестве источника данных для временного периода. Этот параметр доступен, если для *Time period* установлено значение "Widget".
From	Введите или выберите начало временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* установлено значение "Custom".
To	Введите или выберите конец временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* установлено значение "Custom".
Sort entries by	Сортировать записи по: **Time** (по убыванию или возрастанию); **Type** (по убыванию или возрастанию); **Status** (по убыванию или возрастанию); **Recipient** (по убыванию или возрастанию).
Show lines	Укажите, сколько строк журнала действий будет отображаться в виджете.

Часы

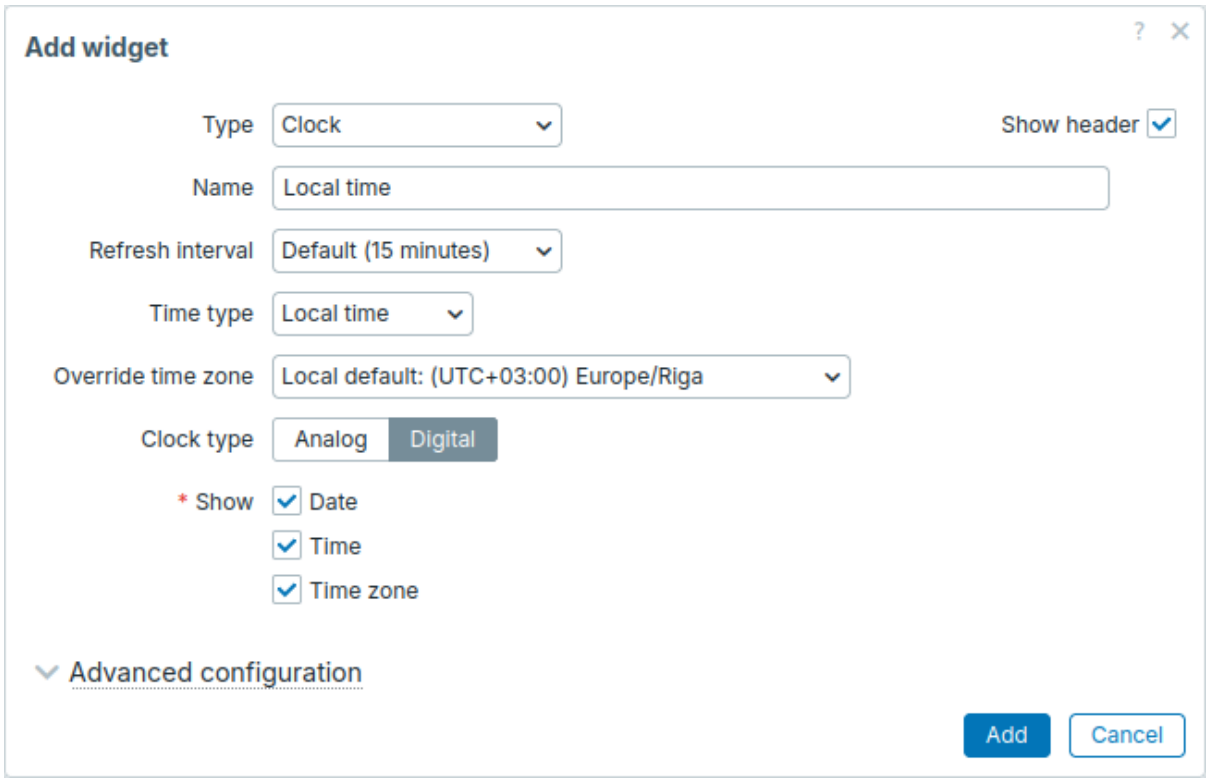
Обзор

Виджет *Clock* отображает местное время, время сервера или узла сети в аналоговом или цифровом формате.



Настройка

Для настройки выберите *Часы* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специфические параметры:

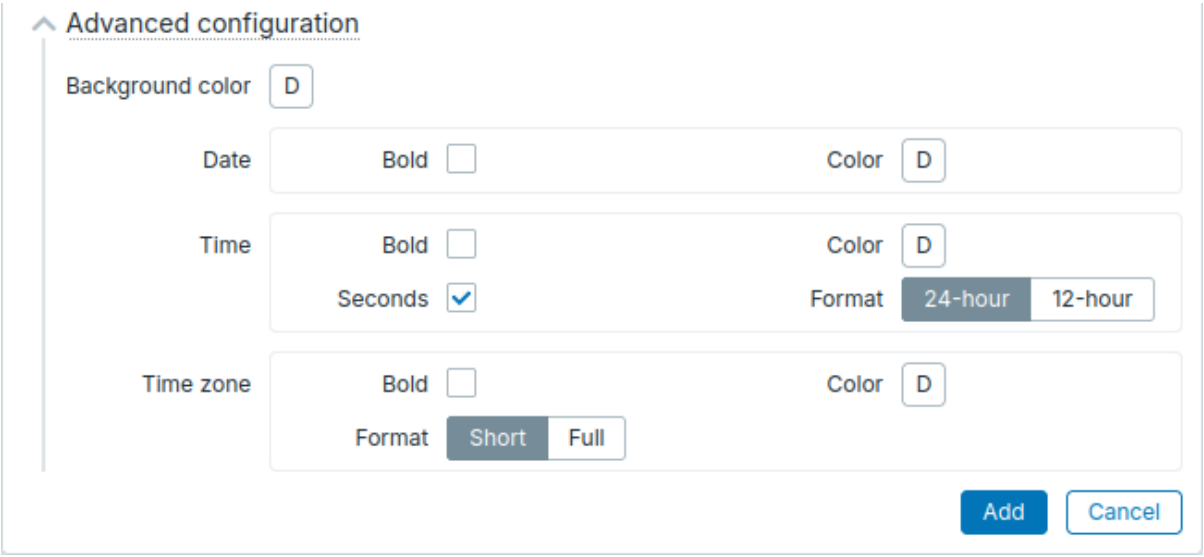
Тип времени	Выберите источник времени: **Местное время** - системное время устройства, на котором запущен браузер; **Время сервера** - часовой пояс, заданный глобально или для пользователя Zabbix; **Время узла сети** - время, полученное из элемента данных узла сети.
Элемент данных	Выберите элемент данных с ключом `system.localtime[local]`, чтобы отображать время узла сети. Этот параметр доступен, если для параметра *Тип времени* установлено значение «Время узла сети».
Тип часов	Выберите тип часов (аналоговые или цифровые).
Переопределить часовой пояс	Выберите часовой пояс. Этот параметр доступен для типов времени *Местное время* и *Время сервера*.

Отображать	Выберите одну или несколько информационных единиц (дату, время, часовой пояс) для отображения на цифровых часах. Размер единиц автоматически адаптируется к отображаемым единицам, размеру виджета и масштабу отображения. Этот параметр доступен, если для параметра *Тип часов* установлено значение «Цифровые».
Расширенная настройка	Нажмите заголовок *Расширенная настройка*, чтобы отобразить параметры расширенной настройки цифровых часов. Этот раздел доступен, если для параметра *Тип часов* установлено значение «Цифровые».

Расширенная настройка

Дополнительные параметры настройки доступны в сворачиваемом разделе *Расширенная настройка* и только для тех элементов, которые выбраны в поле *Отображать* (см. выше).

Кроме того, расширенная настройка позволяет изменить цвет фона всего виджета.



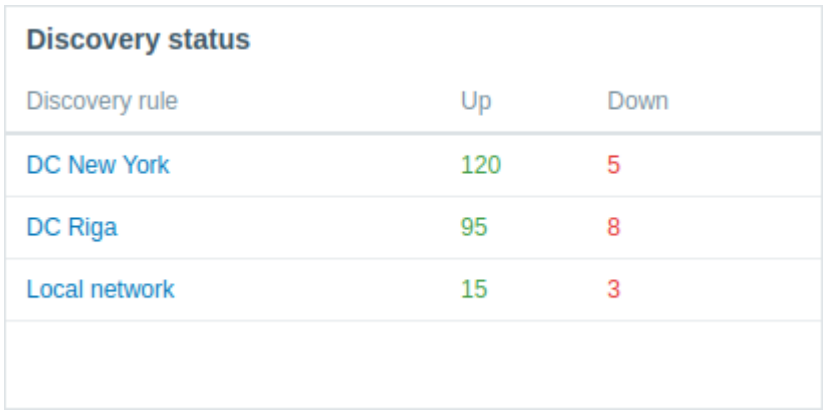
Цвет фона		Выберите цвет фона с помощью палитры. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать значение по умолчанию* в палитре.
Дата		Этот раздел доступен, если для параметра *Отображать* установлен флажок *Дата*.
	Полужирный	Установите флажок, чтобы отображать дату полужирным шрифтом.
	Цвет	Выберите цвет даты с помощью палитры. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать значение по умолчанию* в палитре.
Время		Этот раздел доступен, если для параметра *Отображать* установлен флажок *Время*.
	Полужирный	Установите флажок, чтобы отображать время полужирным шрифтом.
	Цвет	Выберите цвет времени с помощью палитры. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать значение по умолчанию* в палитре.
	Секунды	Установите флажок, чтобы отображать секунды. В противном случае будут отображаться только часы и минуты.
	Формат	Выберите отображение времени в 24- или 12-часовом формате.
Часовой пояс		Этот раздел доступен, если для параметра *Отображать* установлен флажок *Часовой пояс*.
	Полужирный	Установите флажок, чтобы отображать часовой пояс полужирным шрифтом.

Цвет	Выберите цвет часового пояса с помощью палитры. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать значение по умолчанию* в палитре.
Формат	Выберите отображение часового пояса в кратком формате (например, `New York`) или полном формате (например, `(UTC-04:00) America/New York`).

Статус обнаружения

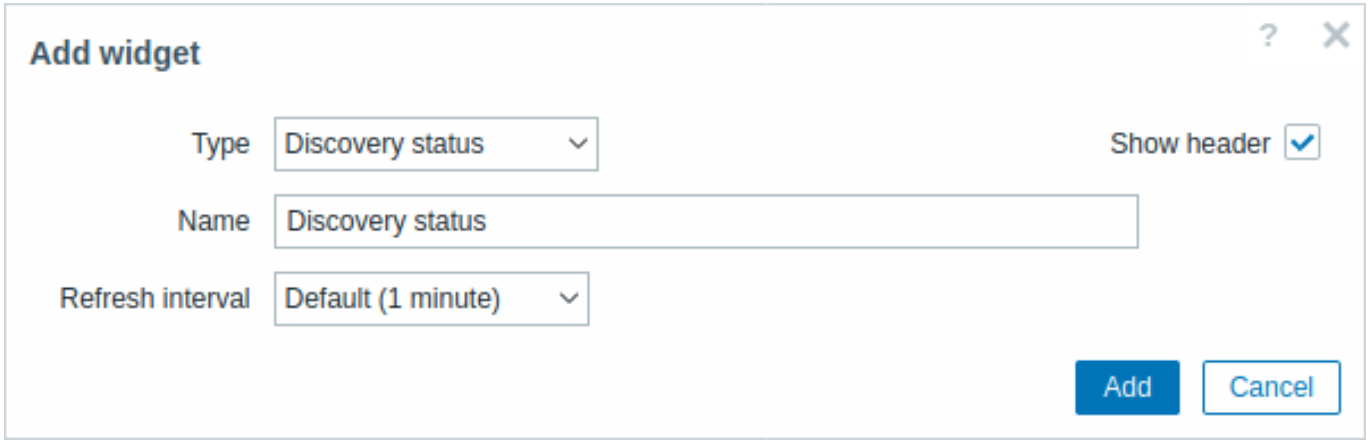
Обзор

Виджет *Discovery status* отображает статус устройств, обнаруженных с помощью включенных правил сетевого обнаружения. Он помогает отслеживать, работает ли сетевое обнаружение корректно, а также количество устройств, которые доступны или недоступны.



Настройка

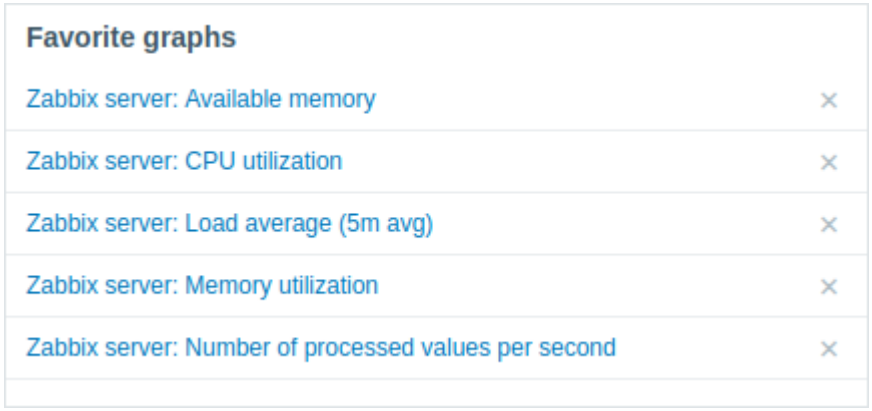
Все параметры настройки этого виджета являются общими для всех виджетов.



Избранные графики

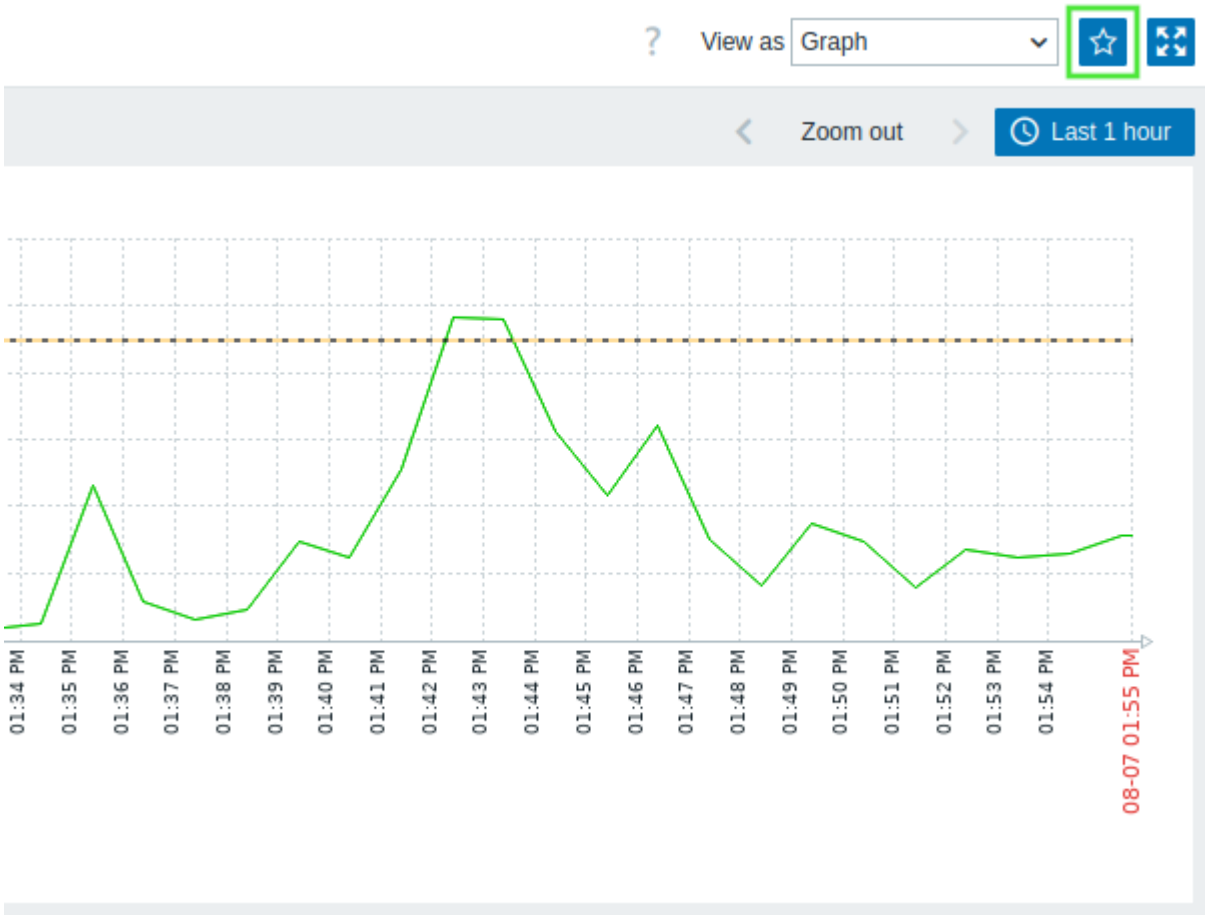
Обзор

Виджет *Избранные графики* отображает алфавитный список ярлыков на графики, отмеченные текущим пользователем как избранные.



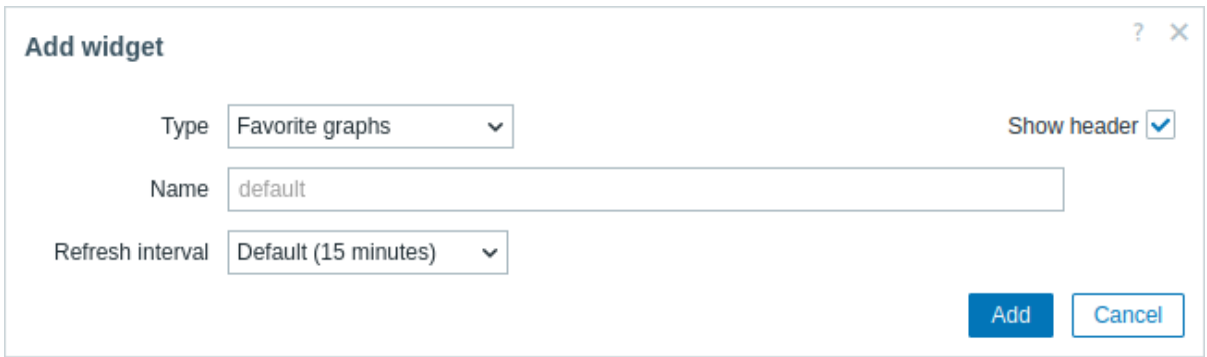
При нажатии x график удаляется из списка и из избранного.

Графики помечаются как избранные при нажатии кнопки  *Добавить в избранное* при просмотре графиков элемента данных в *Мониторинг* > *Последние данные*.



Настройка

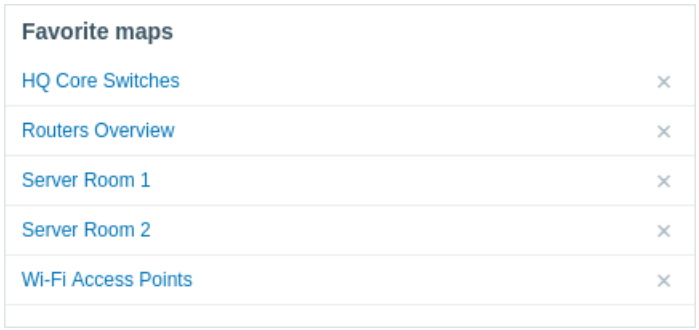
Все параметры настройки этого виджета являются общими для всех виджетов.



Избранные карты

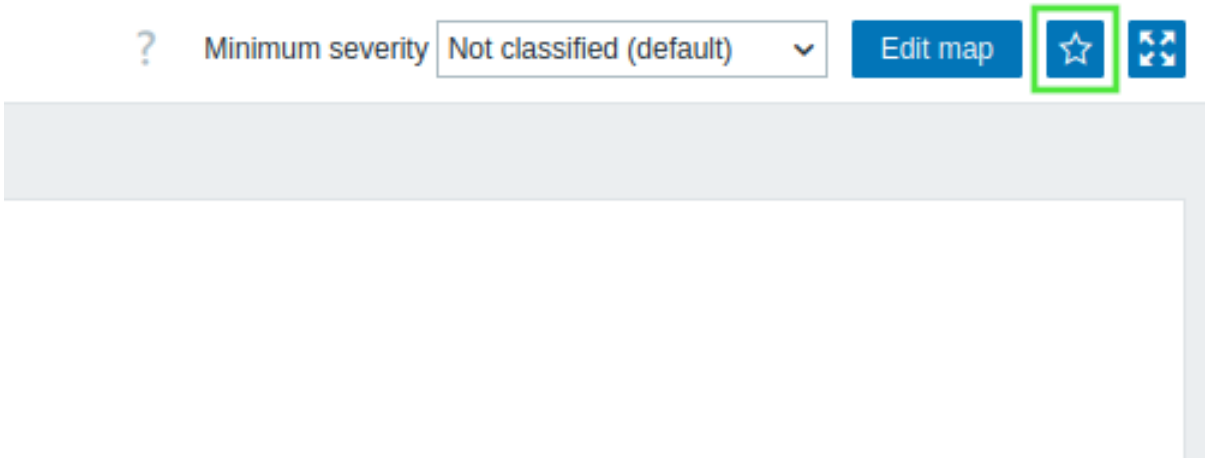
Обзор

Виджет *Избранные карты сети* отображает список ярлыков на карты сети, отмеченные текущим пользователем как избранные, в алфавитном порядке.



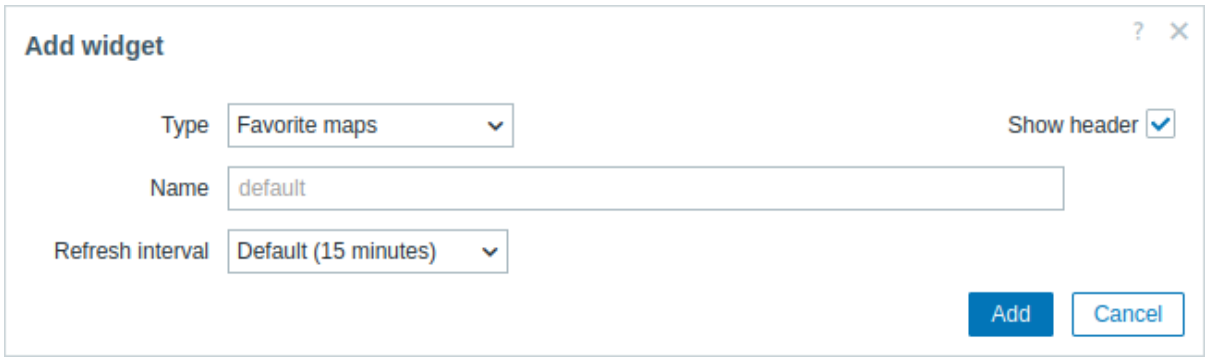
Нажатие x удаляет карту сети из избранного.

Карты сети отмечаются как избранные нажатием кнопки  *Добавить в избранное* при просмотре карт сети в разделе *Monitoring* > *Карты*.



Настройка

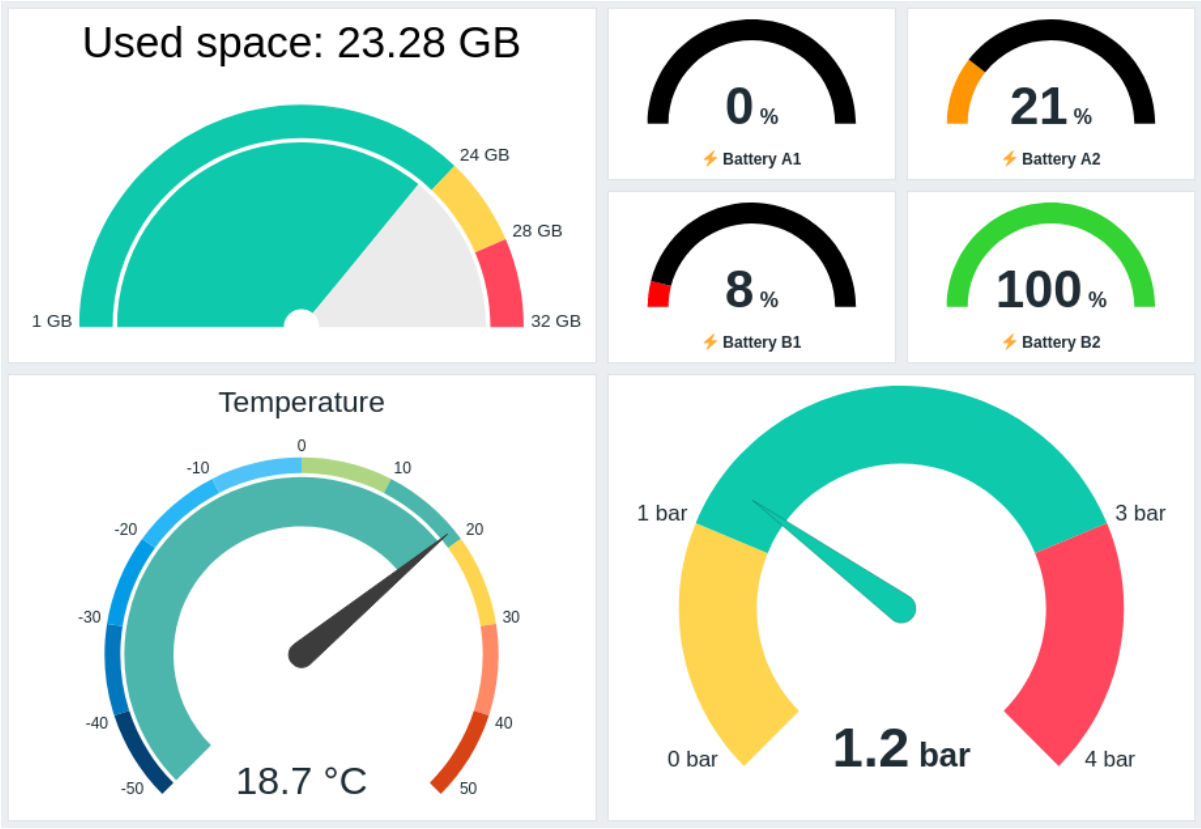
Все параметры настройки этого виджета являются общими для всех виджетов.



Индикатор

Обзор

Виджет *Индикатор* отображает значение одного числового элемента данных в виде индикатора. Он полезен для отслеживания важных метрик, визуализации пороговых значений и обнаружения резких изменений.



Виджет можно настроить для отображения следующих элементов:

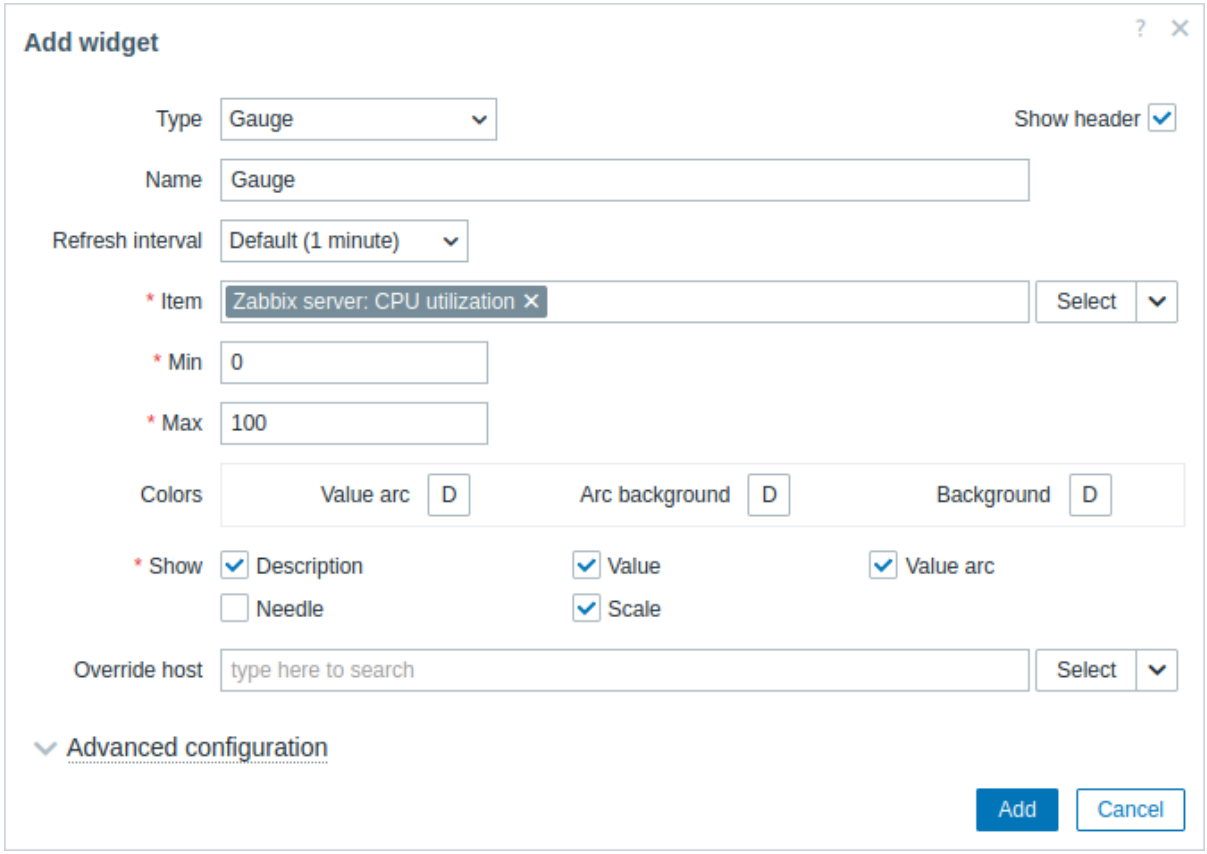
- Описание элемента данных (Используемое пространство: {ITEM.LASTVALUE}, Температура)
- Значение и единицы измерения элемента данных (21 %, 18.7 °C, 1.2 bar)
- Шкала индикатора (1GB/24GB/28GB/32GB, -50/-40/-30/и т. д.)
- Дуга индикатора (дуга значения индикатора и дуга пороговых значений индикатора)
- Стрелка индикатора

При нажатии на виджет открывается простой график для элемента данных.

Информацию, отображаемую в виджете *Индикатор*, можно скачать в виде изображения PNG, выбрав пункт *Download image* в меню *Actions* виджета ..., которое появляется при наведении указателя мыши на виджет. В зависимости от настроек браузера изображение либо автоматически сохраняется в папке `Downloads`, либо открывается диалоговое окно *Save*, в котором можно выбрать расположение файла.

Конфигурация

Чтобы настроить, выберите *Gauge* в качестве типа:



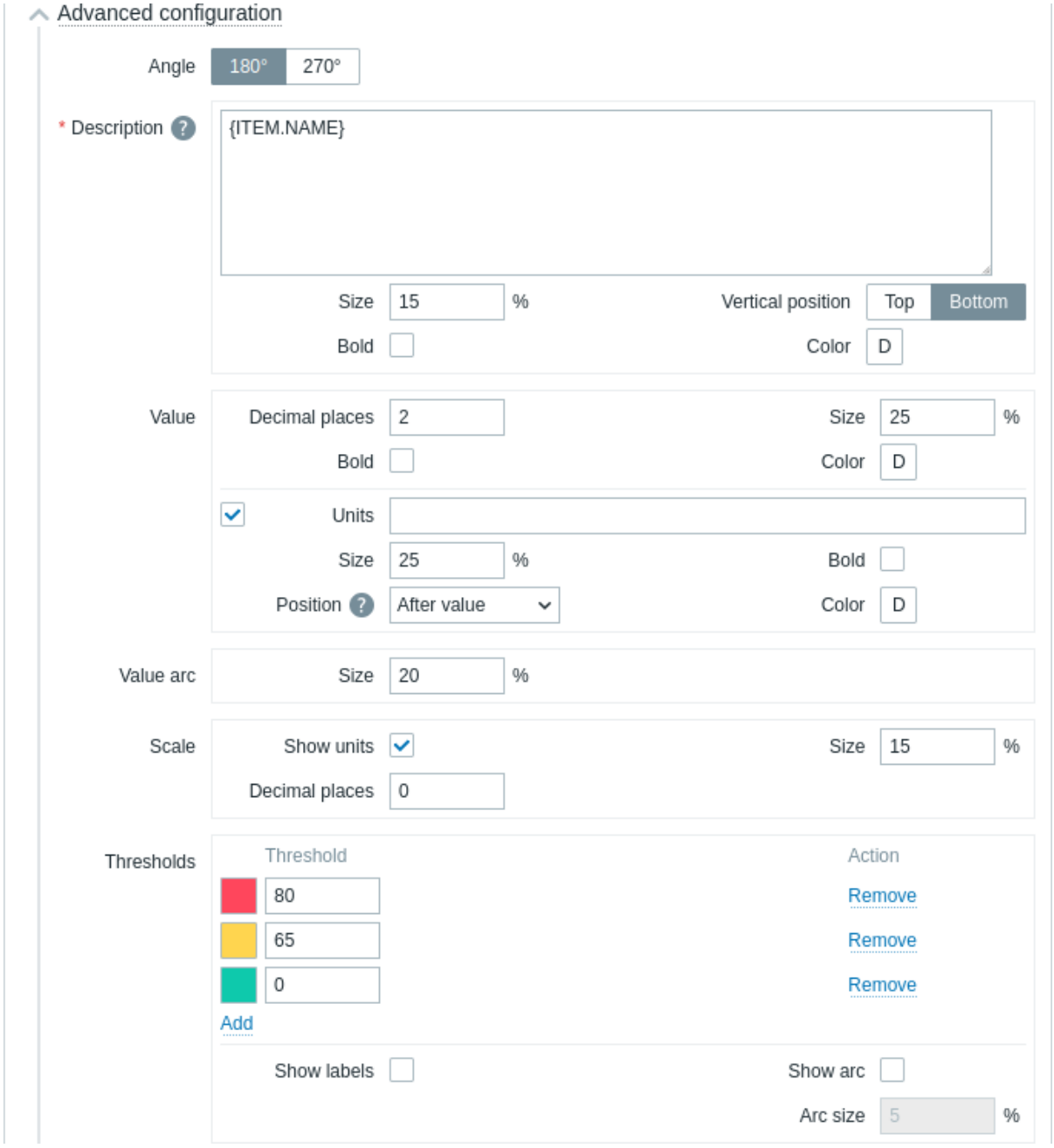
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Item	Выберите элемент данных. Либо выберите совместимый виджет в качестве источника данных для элементов данных. Это поле поддерживает автодополнение, поэтому при вводе имени элемента данных будет предлагаться выпадающий список совпадающих элементов данных. Обратите внимание, что можно выбрать только элементы данных, которые возвращают числовые данные (кроме двоичных данных).
Min	Введите минимальное значение шкалы. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Max	Введите максимальное значение шкалы. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Colors	Выберите цвет в палитре: **Value arc** - выберите цвет дуги значения шкалы; **Arc background** - выберите цвет фона дуги значения шкалы и дуги порогов шкалы; **Background** - выберите цвет фона виджета. ”D” означает цвет по умолчанию, который зависит от темы веб-интерфейса. Если заданы *Thresholds*, цвет по умолчанию для **Value arc** зависит от цвета порога. Чтобы вернуться к цвету по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Show	Установите флажок, чтобы отображать соответствующий элемент шкалы - описание, значение, дугу значения, стрелку, шкалу (минимальное и максимальное значение шкалы в начале и конце дуги шкалы). Снимите флажок, чтобы скрыть. Должен быть выбран как минимум один элемент. Обратите внимание, что стрелка шкалы и шкала могут отображаться, если отображается дуга значения шкалы или дуга порогов шкалы (см. параметры расширенной настройки). Также обратите внимание, что если отображается стрелка шкалы, значение размещается под стрелкой; если стрелка скрыта, значение выравнивается по нижнему краю дуги шкалы.
Override host	Выберите совместимый виджет или селектор узла сети на панели мониторинга в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели мониторинга.

Advanced configuration	Нажмите заголовок *Advanced configuration*, чтобы отобразить параметры расширенной настройки. Здесь же можно настроить элементы шкалы, выбранные в поле *Show*.

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка (Advanced configuration)*:



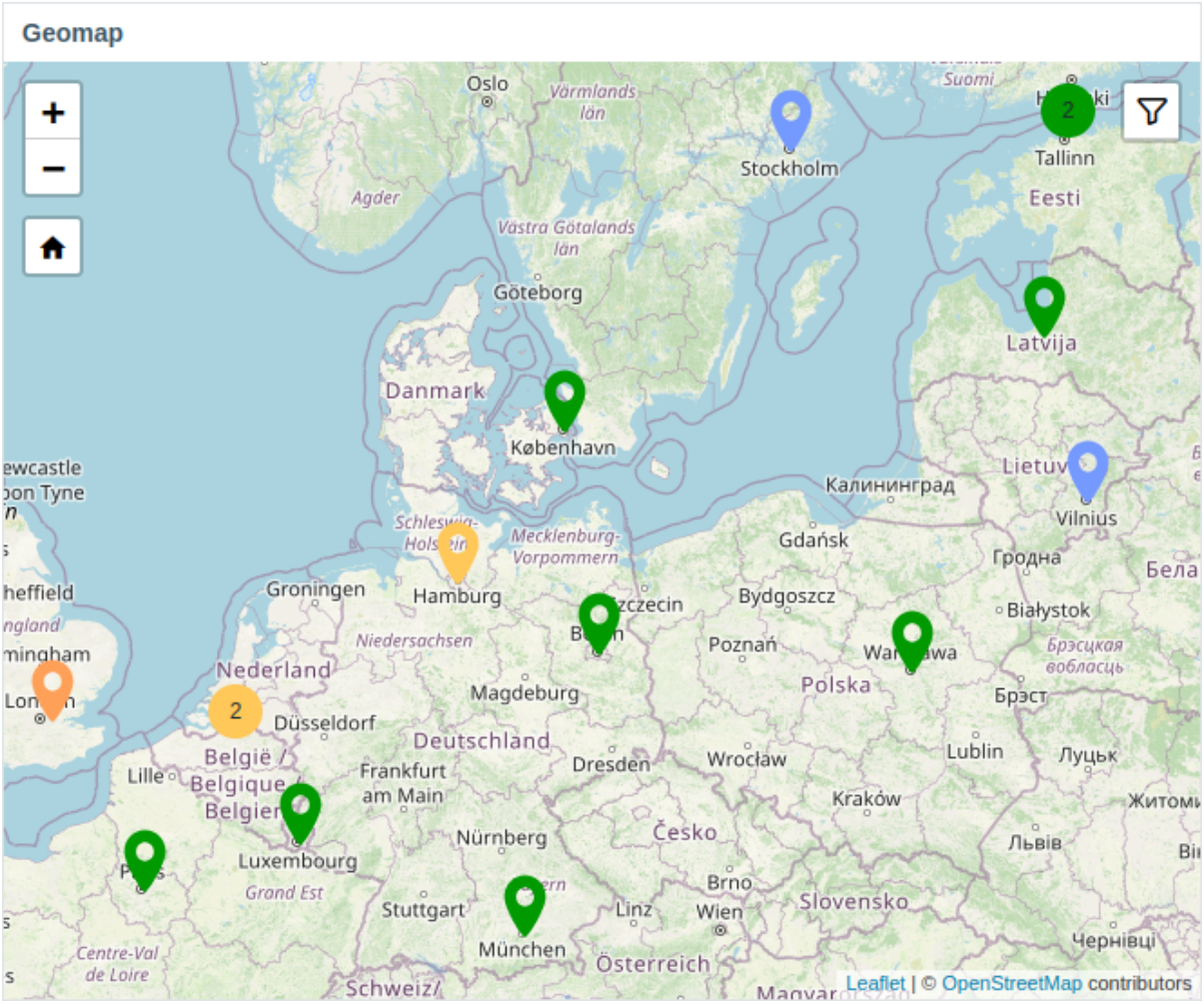
Угол (Angle)	Выберите угол шкалы (180° или 270°).
Описание (Description)	
Описание (Description)	Введите описание элемента данных. Это описание может переопределять название элемента данных по умолчанию. Поддерживаются многострочные описания. Допустима комбинация текста и поддерживаемых макросов. Поддерживаются макросы {HOST.*}, {ITEM.*}, {INVENTORY.*} и пользовательские макросы.
Размер (Size)	Введите высоту шрифта для описания элемента данных (в процентах, относительно общей высоты виджета).
Расположение по вертикали (Vertical position)	Выберите расположение по вертикали описания элемента данных (по верхнему либо нижнему краю, относительно дуги шкалы).

Жирный (Bold)	Установите флажок, чтобы описание отображалось жирным шрифтом.
Цвет (Color)	Выберите из палитры цветов цвет описания элемента данных. «D» означает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуться к значению по умолчанию, нажмите в палитре кнопку *Использовать умолчание (Use default)*.
Значение (Value)	
Число десятичных знаков (Decimal places)	Введите количество десятичных знаков, отображаемых со значением. Эта опция влияет только на элементы данных, которые возвращают значения с типом числовой (с плавающей точкой).
Размер (Size)	Введите высоту шрифта для значения (в процентах, относительно высоты дуги).
Жирный (Bold)	Установите флажок, чтобы значение отображалось жирным шрифтом.
Цвет (Color)	Выберите из палитры цветов цвет значения. «D» означает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуться к значению по умолчанию, нажмите в палитре кнопку *Использовать умолчание (Use default)*.
Единицы измерения (Units)	
Единицы измерения (Units)	Установите флажок, чтобы вместе со значением элемента данных отображались единицы измерения. Если вы вводите название единицы измерения, это переопределит единицы измерения, заданные в настройках элемента данных.
Размер (Size)	Введите высоту шрифта для единиц измерения элемента данных (в процентах, относительно высоты дуги).
Жирный (Bold)	Установите флажок, чтобы единицы измерения отображались жирным шрифтом.
Расположение (Position)	Выберите расположение единиц измерения элемента данных (выше, ниже, перед или после, относительно значения элемента данных). Эта опция игнорируется для слудещих единиц измерения, относящихся ко времени: unixtime, uptime, s.
Цвет (Color)	Выберите из палитры цветов цвет единиц измерения. «D» означает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуться к значению по умолчанию, нажмите в палитре кнопку *Использовать умолчание (Use default)*.
Значение дуги (Value arc)	
Размер дуги (Arc size)	Введите высоту значения дуги радиального датчика (в процентах, относительно радиуса дуги).
Стрелка (Needle)	
Цвет (Color)	Выберите из палитры цветов цвет стрелки радиального датчика. «D» означает цвет по умолчанию (зависит от темы веб-интерфейса). Если заданы *Пороги*, цвет по умолчанию для стрелки зависит от цвета порога . Чтобы вернуться к значению по умолчанию, нажмите в палитре кнопку *Использовать умолчание (Use default)*.
Шкала (Scale)	
Показывать единицы измерения (Show units)	Установите флажок, чтобы вместе с минимальным и максимальным значением шкалы отображались единицы измерения.
Размер (Size)	Введите высоту шрифта для минимального и максимального значения шкалы (в процентах, относительно высоты дуги шкалы).
Число десятичных знаков (Decimal places)	Введите количество десятичных знаков, отображаемых с минимальным и максимальным значением шкалы. Эта опция влияет только на элементы данных, которые возвращают значения с типом числовой (с плавающей точкой).
Пороги (Thresholds)	
Пороги (Thresholds)	Нажмите *Добавить (Add)* для добавления порога, выберите цвет порога из палитры и укажите числовое значение. При сохранении список порогов будет отсортирован в возрастающем порядке. Обратите внимание, что цвета, настроенные для порогов, будут правильно отображаться только для числовых элементов данных. Поддерживаются суффиксы (например, «1d», «2w», «4K», «8G»). Поддерживаются преобразования значений.
Показывать подписи (Show labels)	Установите флажок, чтобы пороговые значения отображались как подписи на шкале.
Показать дугу (Show arc)	Установите флажок, чтобы отображалась дуга порогов.
Размер дуги (Arc size)	Введите высоту дуги порогов (в процентах, относительно радиуса дуги значений).

Геокарта

Обзор

Виджет *Geomap* отображает узлы сети в виде маркеров на интерактивной карте на основе Leaflet, при этом цвета маркеров указывают на наиболее серьезную проблему каждого узла сети.

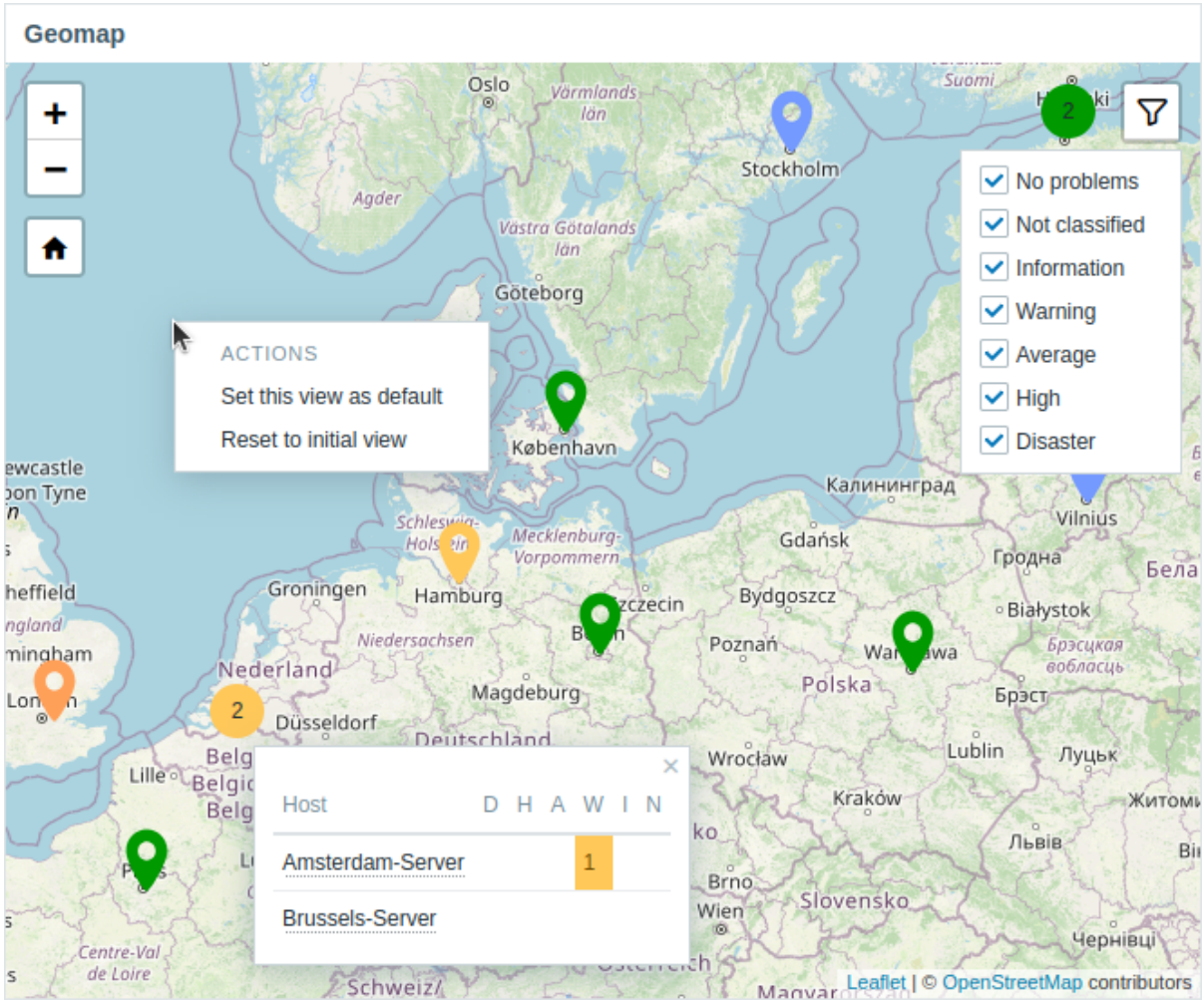


Note:
Zabbix предлагает несколько предопределенных поставщиков тайлов карты и позволяет добавлять собственные в *Administration* > *General* > *Geographical maps*.

По умолчанию виджет отображает все включенные узлы сети, у которых в инвентаре заданы корректные координаты (широта от -90 до 90, долгота от -180 до 180).

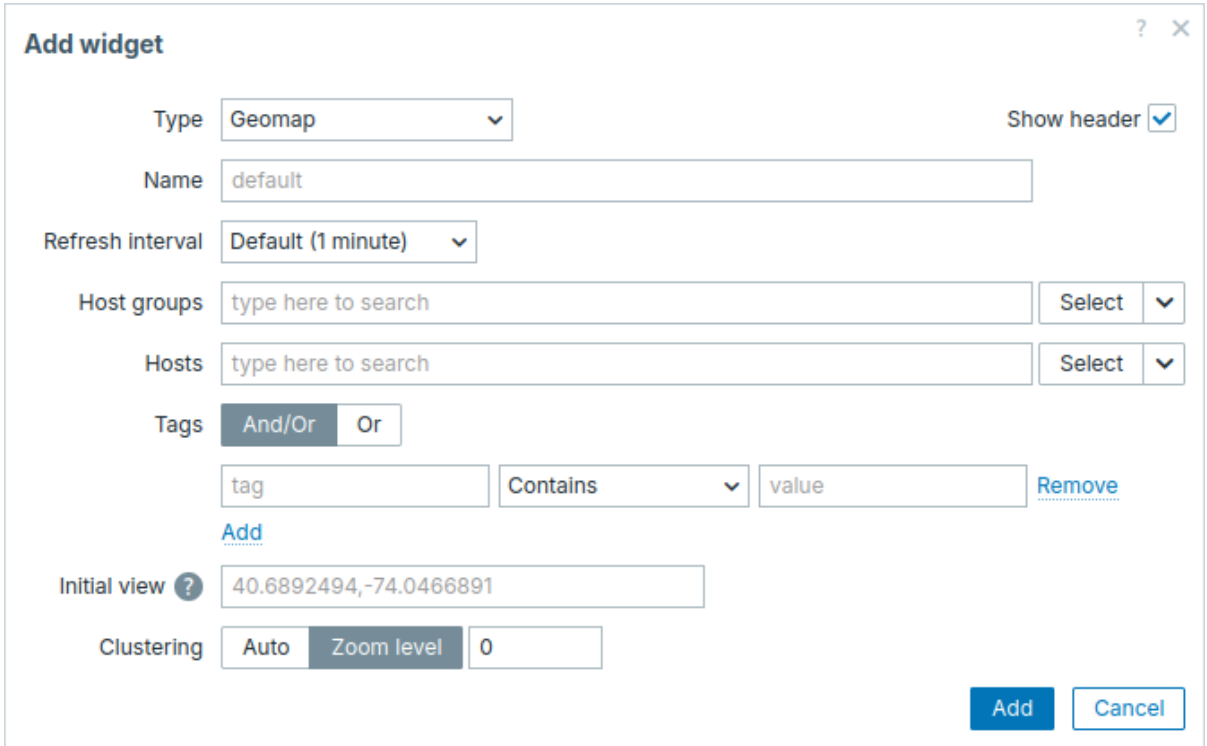
Для взаимодействия с виджетом используйте следующие параметры:

- Щелчок по маркеру показывает имя узла сети и нерешенные проблемы; щелчок по имени узла сети открывает меню узла сети.
- Фильтруйте узлы сети по степени серьезности проблемы с помощью значка фильтра в правом верхнем углу.
- Щелкните правой кнопкой мыши по карте, чтобы задать или сбросить вид по умолчанию; используйте значок домика, чтобы вернуться к виду по умолчанию.



Конфигурация

Чтобы добавить виджет, выберите *Geomap* в качестве типа.



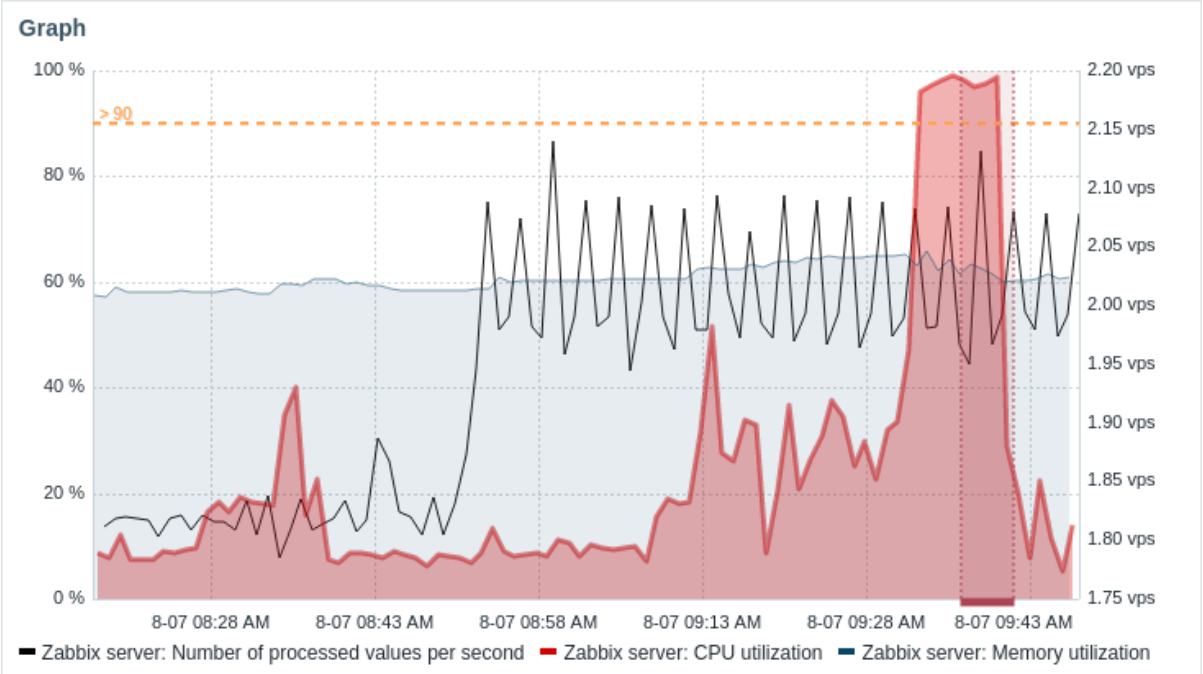
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети, которые будут отображаться на карте. В качестве альтернативы выберите совместимый виджет как источник данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Если не выбрано ничего ни в полях *Группы узлов сети*, ни в *Узлы сети*, будут отображены все узлы сети с допустимыми координатами. Этот параметр недоступен при настройке виджета на шаблонной панели.
Узлы сети	Выберите узлы сети, которые будут отображаться на карте. В качестве альтернативы выберите совместимый виджет или панель как источник данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Если не выбрано ничего ни в полях *Группы узлов сети*, ни в *Узлы сети*, будут отображены все узлы сети с допустимыми координатами. Этот параметр недоступен при настройке виджета на шаблонной панели.
Теги	Укажите теги, чтобы ограничить количество узлов сети, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия. Этот параметр недоступен при настройке виджета на шаблонной панели.
Начальный вид	Координаты центра, разделенные запятыми, и необязательный уровень масштабирования, который будет отображаться при первоначальной загрузке виджета, в формате `<latitude>,<longitude>,<zoom>` Если указан начальный масштаб, виджет Geomap загружается с заданным уровнем масштабирования. В противном случае начальный масштаб вычисляется как половина от максимального масштаба для конкретного поставщика тайлов. Начальный вид игнорируется, если задан вид по умолчанию (см. ниже). Примеры: 40.6892494,-74.0466891 40.6892494,-122.0466891
Кластеризация	Укажите, как близко расположенные маркеры узлов сети объединяются в один маркер с указанием количества: **Auto** - маркеры объединяются автоматически; **Zoom level** - маркеры объединяются на основе указанного уровня масштабирования (0 - максимальное уменьшение, 30 - максимальное увеличение). Например, если параметр установлен в 0, маркеры не объединяются ни на одном уровне масштабирования; если установлен в 15, маркеры объединяются, начиная с максимального уменьшения и до уровня масштабирования 15; если установлен в 30, маркеры объединяются на всех уровнях масштабирования. Обратите внимание, что максимальный уровень увеличения определяется поставщиком тайлов карты.

График

Обзор

Виджет *График* отображает числовые данные элементов данных в виде векторного графика. Он помогает отслеживать тенденции, выявлять проблемы и сравнивать значения с течением времени и между узлами сети.



При наведении указателя мыши на график отображается всплывающая подсказка со списком элементов данных, отсортированных по значению в порядке убывания.

При наведении указателя мыши на элемент данных во всплывающей подсказке выделяется его график, а остальные графики становятся менее заметными; выбор элемента данных передает его данные другим виджетам.

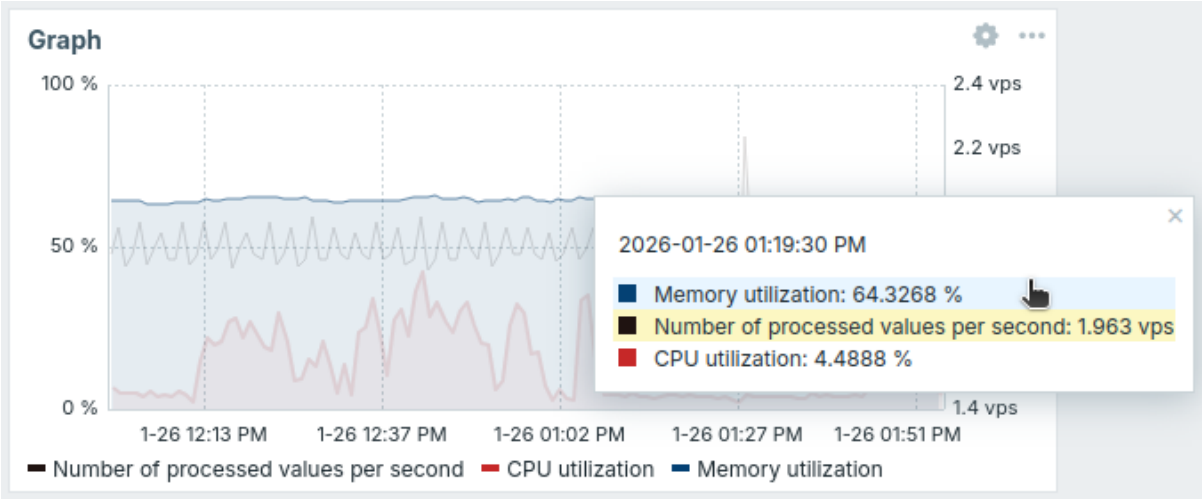
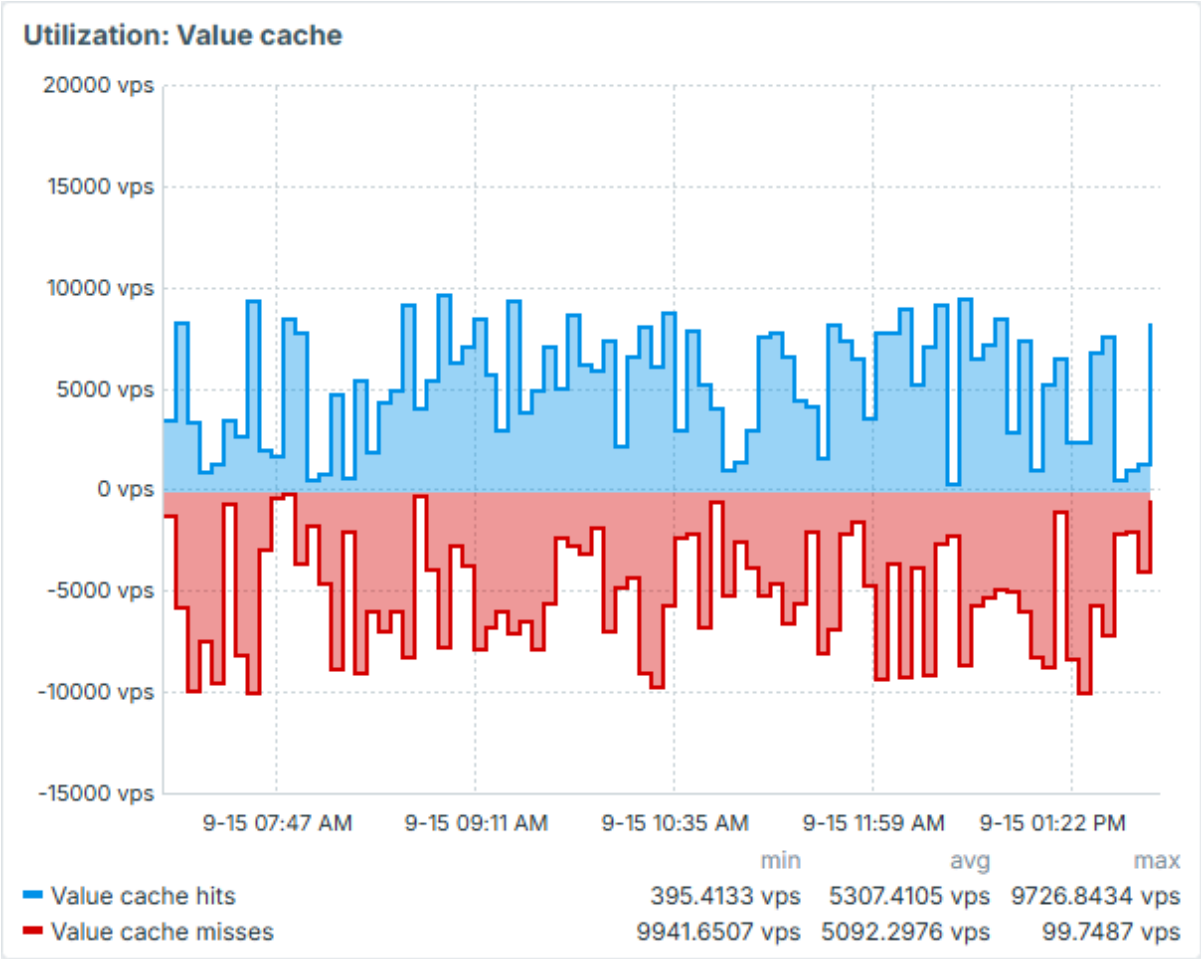


График можно отображать с инвертированными значениями оси Y; это влияет только на визуальное представление и не изменяет исходные значения.

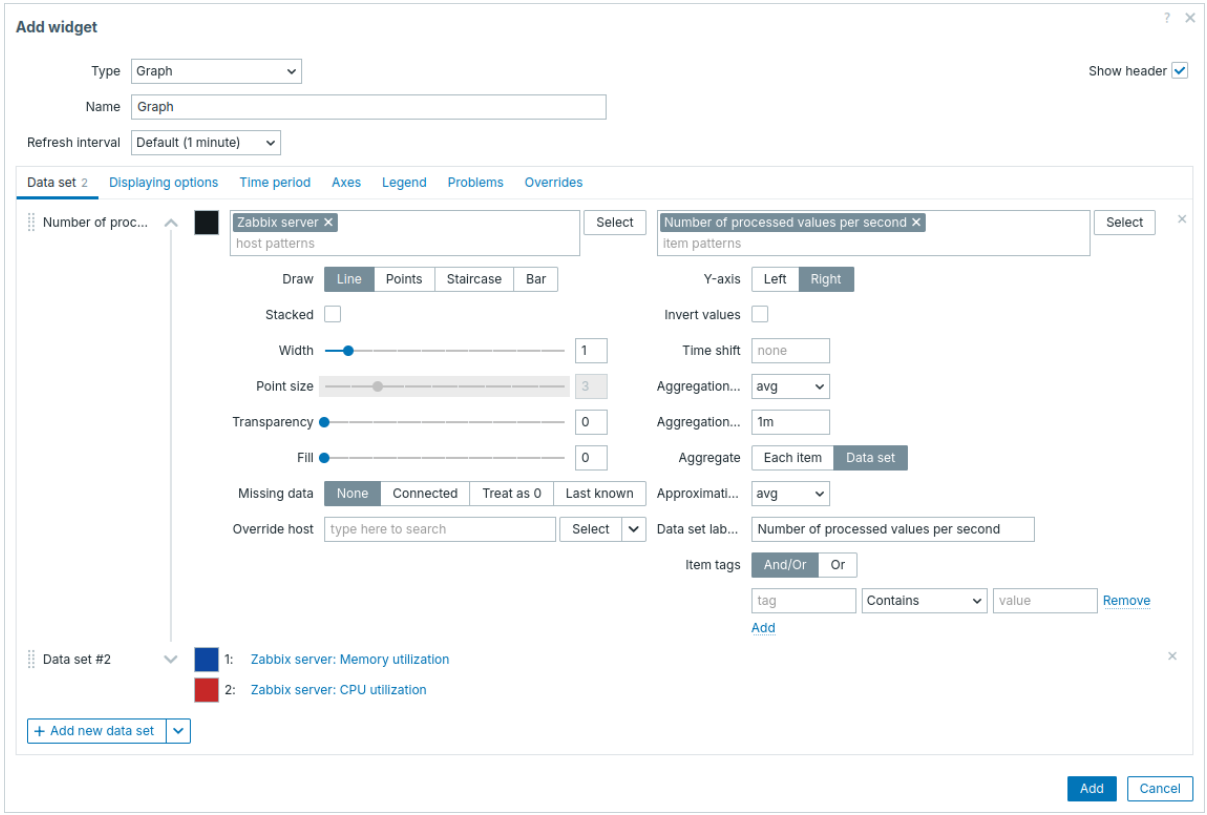


Также доступен устаревший виджет График (классический).

Информацию, отображаемую в виджете *График*, можно скачать в виде изображения PNG, выбрав пункт *Скачать изображение* в меню *Действия* виджета ..., которое появляется при наведении указателя мыши на виджет. В зависимости от настроек браузера изображение автоматически сохраняется в папке `Downloads` или открывается диалоговое окно *Сохранить*, в котором можно выбрать расположение файла.

Настройка

Для настройки выберите *График (Graph)* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбирать данные для графика путем добавления наборов данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* - отображаются данные соответствующих элементов данных. Можно выбрать один базовый цвет или строку палитры, чтобы назначить отдельные цвета каждому соответствующему элементу данных.
- *Список элементов данных* - отображаются данные выбранных элементов данных. Цвет каждого элемента данных можно выбрать отдельно с помощью палитры.

По умолчанию добавляется набор данных типа *Шаблоны элементов данных*.

Набор данных	Для набора данных **Шаблоны элементов данных**: Выберите или введите шаблоны узлов сети и элементов данных; на графике будут отображаться данные элементов данных, соответствующих этим шаблонам; можно отображать до 50 элементов данных. Для выбора можно использовать шаблоны с подстановочными символами (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным символом, введите строку вручную и нажмите *Enter*. Символ подстановки всегда интерпретируется, поэтому невозможно добавить, например, элемент данных с именем *item** отдельно, если существуют другие соответствующие элементы данных (например, *item2*, *item3*). Для наборов данных типа ”Шаблоны элементов данных” обязательно указать шаблоны узлов сети и элементов данных. См. также: Подробные сведения о настройке набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для графика, нажав кнопку *Добавить элемент данных*. Также можно выбрать совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*. Для наборов данных типа ”Список элементов данных” обязательно указать элементы данных или виджеты. См. также: Подробные сведения о настройке набора данных. Обратите внимание, что разрешены только числовые типы элементов данных. При настройке виджета на панели шаблона параметр для указания шаблонов узлов сети недоступен, а параметр для указания списка элементов данных позволяет выбирать только элементы данных, настроенные в шаблоне.
Отображение	Выберите тип отображения метрики. Доступные типы отображения: *Линия* (установлен по умолчанию), *Точки*, *Ступенчатая линия* и *Столбцы*. Обратите внимание, что если на графике с линией или ступенчатой линией имеется только одна точка данных, она отображается как точка независимо от выбранного типа отображения. Размер точки рассчитывается на основе ширины линии, но не может быть меньше 3 пикселей, даже если ширина линии меньше.
С накоплением	Установите флажок, чтобы отображать данные с накоплением (с заполненными областями). Этот параметр отключен, если выбран тип отображения *Точки*.
Ширина	Укажите ширину линии. Этот параметр доступен, если выбран тип отображения *Линия* или *Ступенчатая линия*.
Размер точки	Укажите размер точки. Этот параметр доступен, если выбран тип отображения *Точки*.
Прозрачность	Укажите уровень прозрачности.
Заполнение	Укажите уровень заполнения. Этот параметр доступен, если выбран тип отображения *Линия* или *Ступенчатая линия*.
Отсутствующие данные	Выберите способ отображения отсутствующих данных: **Нет** - промежуток остается пустым; **Соединять** - соединяются два граничных значения; **Считать равными 0** - отсутствующие данные отображаются как значения 0; **Последнее известное** - отсутствующие данные отображаются с тем же значением, что и последнее известное значение; не применяется к типам отображения *Точки* и *Столбцы*.
Переопределить узел сети	Выберите совместимый виджет или селектор узлов сети панели в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Ось Y	Выберите сторону графика, на которой будет отображаться ось Y.

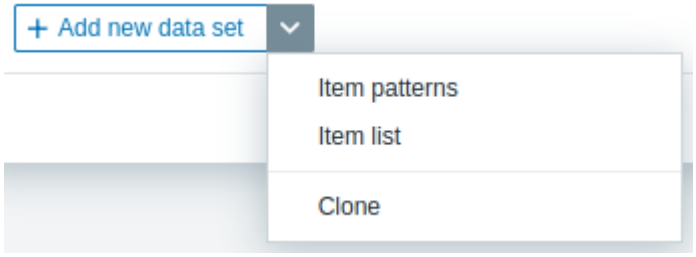
Инвертировать значения	Установите флажок, чтобы перед отображением на графике умножить все значения оси Y на -1. Это влияет только на визуальное представление; исходные данные остаются без изменений. Этот параметр не влияет на всплывающую подсказку, легенду или линию процентиля. Если для оси выбран параметр *Простые триггеры*, пороговые значения триггеров будут отображаться инвертированными.
Смещение времени	При необходимости укажите смещение времени. В этом поле можно использовать суффиксы времени. Допускаются отрицательные значения.
Функция агрегации	Укажите функцию агрегации, которая будет использоваться для каждого элемента данных или всего набора данных в пределах выбранного *Интервала агрегации*: **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **sum** - отображать сумму значений; **count** - отображать количество значений; **first** - отображать первое значение; **last** - отображать последнее значение. Если выбрано **не используется**, отображаются все значения (без агрегации). Для *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для *count* нечисловые данные преобразуются в числовые. См. также: Агрегация на графиках.
Интервал агрегации	Укажите интервал агрегации значений. В этом поле можно использовать суффиксы времени. Числовое значение без суффикса считается количеством секунд. Обратите внимание, что если виджет настроен на отображение исторических данных на основе трендов (*Выбор данных истории* установлен в *Тренды* или *Авто*), рекомендуется использовать интервал агрегации, кратный 1 часу (например, 3600, 60m, 1h, 3h и т. д.). Тренды хранят агрегированные значения с интервалом в 1 час, поэтому использование интервала агрегации, не кратного 1 часу (например, 100s, 7min, 15min, 90min и т. д.), может привести к результатам, которые сложно интерпретировать.
Агрегировать	Укажите, следует ли выполнять агрегацию: **Каждый элемент данных** - каждый элемент данных в наборе агрегируется и отображается отдельно; **Набор данных** - все элементы данных набора агрегируются и отображаются как одно значение.
Аппроксимация	Укажите, какое значение отображать, если на один вертикальный пиксель графика приходится более одного значения: **all** - отображать наименьшее, наибольшее и среднее значения; **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение. Этот параметр полезен при отображении графика за большой период времени с частым интервалом обновления (например, значений за один год, собранных каждые 10 минут).
Метка набора данных	Укажите метку набора данных, отображаемую в настройках графика *Набор данных* и в *Легенде* графика (для агрегированных наборов данных). Все наборы данных нумеруются, включая наборы с указанной *Меткой набора данных*. Если метка не указана, набору данных автоматически присваивается метка в соответствии с его номером (например, ”Набор данных №2”, ”Набор данных №3” и т. д.). Нумерация наборов данных пересчитывается после изменения их порядка или перетаскивания. Слишком длинные метки наборов данных сокращаются, чтобы соответствовать месту отображения (например, ”Number of proc...”).

Теги элементов данных	Укажите теги для фильтрации элементов данных, отображаемых в виджете. Можно включать или исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (сопоставление подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (сопоставление подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега группируются с помощью условия ИЛИ; **ИЛИ** - достаточно выполнения одного условия.

Подробные сведения о настройке наборов данных

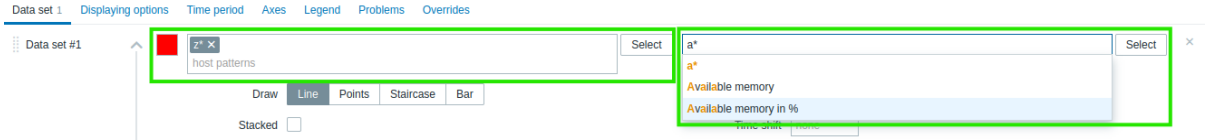
Существующие наборы данных отображаются в списке. Вы можете:

- Нажать значок перемещения и перетащить набор данных на новое место в списке.
- Нажать значок разворачивания, чтобы развернуть сведения о наборе данных. При разворачивании этот значок превращается в значок сворачивания.
- Нажать значок цвета, чтобы открыть палитру. Можно ввести шестнадцатеричный код, выбрать сплошной образец цвета или перейти на вкладку *Palette* и выбрать строку предопределенных цветов. Выбранный цвет применяется непосредственно к наборам данных *Item list* или используется как базовый для создания оттенков в наборах данных *Item patterns*. Используйте клавишу Tab для перехода между элементами управления диалогового окна, клавиши со стрелками для навигации по образцам цветов или строкам палитры, клавишу Enter для выбора и клавишу Esc для отмены.
- Нажать кнопку *Add new data set*, чтобы добавить пустой набор данных, в котором можно выбрать узел сети и шаблоны элементов данных. Если нажать значок со стрелкой вниз рядом с кнопкой *Add new data set*, появится раскрывающееся меню, позволяющее добавить новый набор данных *Item patterns* или *Item list*, а также *Clone* текущего открытого набора данных. Если все наборы данных свернуты, параметр *Clone* недоступен.

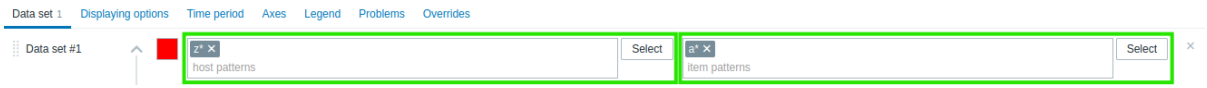


Набор данных **Item patterns** содержит поля *Host patterns* и *Item patterns*, которые распознают как полные имена, так и шаблоны, содержащие символ подстановки (*). Эта функция позволяет выбрать все имена узлов сети и имена элементов данных, содержащие выбранный шаблон. При вводе имени элемента данных или шаблона элемента данных в поле *Item patterns* в раскрывающемся списке отображаются только элементы, относящиеся к выбранному имени(ам) узла сети.

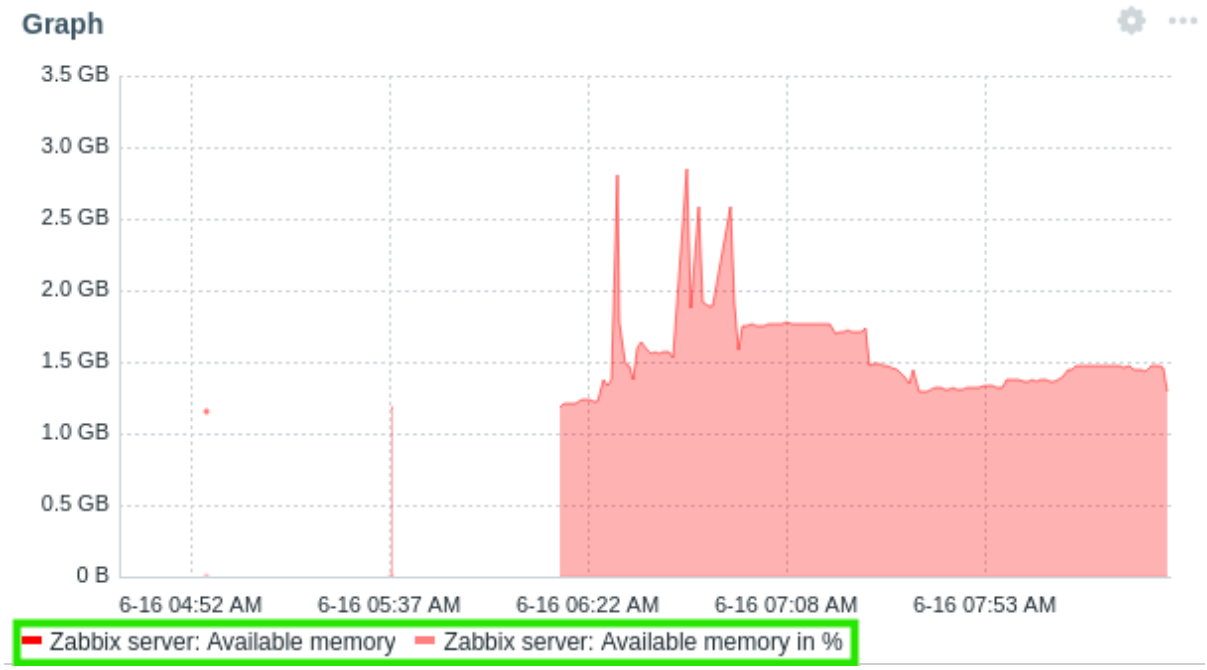
Например, если в поле *Host patterns* ввести шаблон **z***, в раскрывающемся списке будут показаны все имена узлов сети, содержащие этот шаблон: **z*, Zabbix server, Zabbix proxy**. После нажатия *Enter* этот шаблон принимается и отображается как **z***. Аналогично, если в поле *Item patterns* ввести шаблон **a***, в раскрывающемся списке будут показаны все имена элементов данных, содержащие этот шаблон: **a*, Available memory, Available memory in %**.



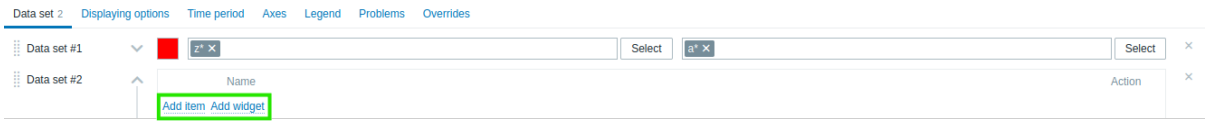
После нажатия *Enter* шаблон принимается и отображается как **a***.



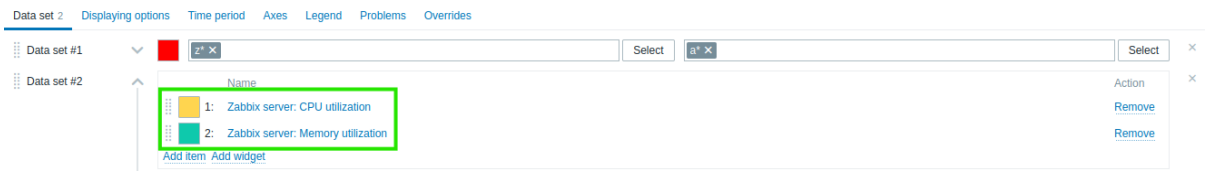
Затем график отображает все элементы данных, относящиеся к выбранному имени(ам) узла сети.



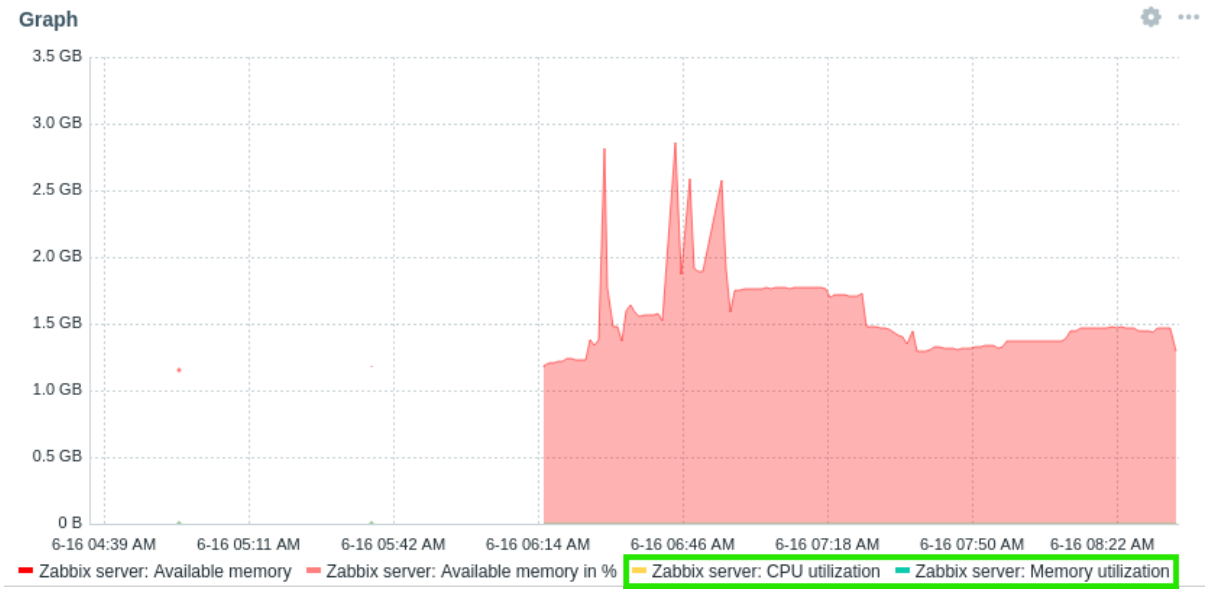
Набор данных **Список элементов данных** содержит кнопку *Добавить элемент данных*, которая позволяет добавлять элементы данных для отображения на графике. Вы также можете добавлять совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*.



Например, при нажатии кнопки *Добавить элемент данных* открывается всплывающее окно, содержащее параметр *Узел сети*. После выбора узла сети в списке отображаются все его элементы данных, доступные для выбора.



После выбора одного или нескольких элементов данных они будут отображены в списке элементов данных набора данных и на графике.

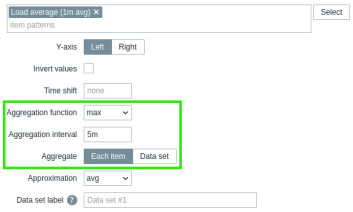


Агрегация на графиках

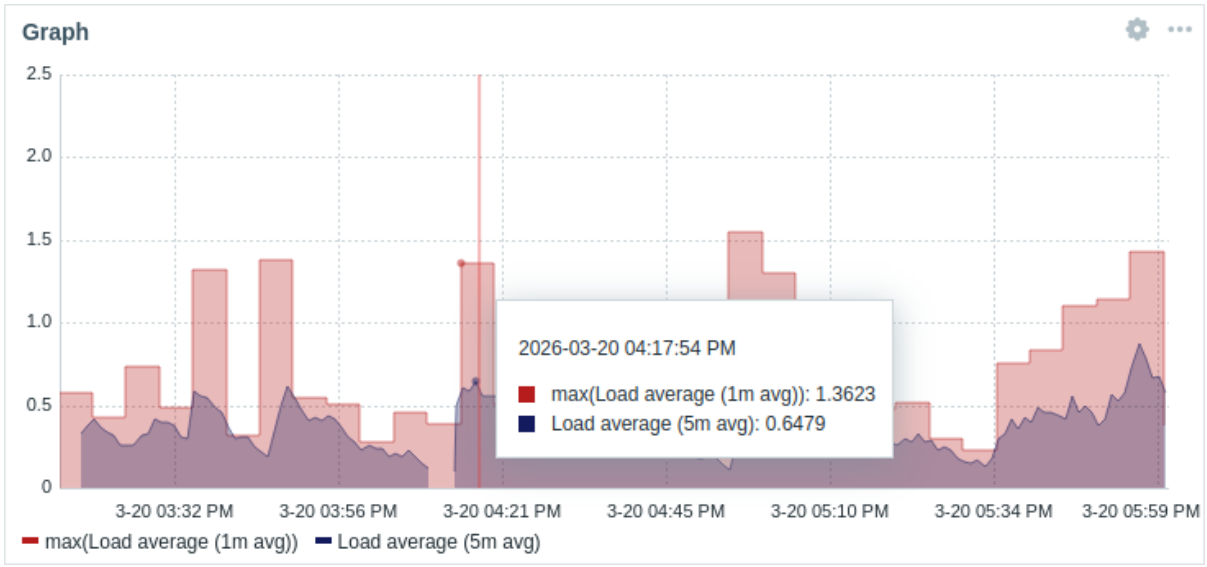
Агрегация позволяет отображать агрегированное значение (минимальное, максимальное, среднее и т. д.) за указанный интервал (5 минут, час, день и т. д.) вместо всех отдельных значений. В наборе данных агрегация может применяться либо к каждому элементу данных отдельно, либо ко всему набору данных.

Примеры использования функции агрегации: - Среднее количество запросов к серверу NGINX за день. - Наименьший объем свободного дискового пространства среди кластеров за неделю.

В следующем примере к элементу данных применяется функция агрегации "max" с интервалом агрегации пять минут:



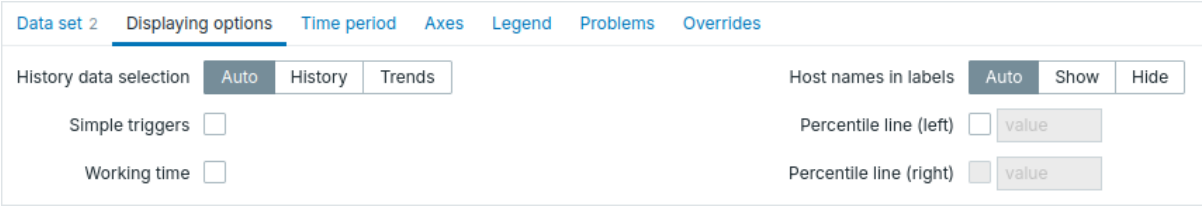
В примере показана максимальная загрузка ЦП за пять минут (агрегированная из средних значений за одну минуту) в сравнении со средней загрузкой ЦП за пять минут.



Если для легенды выбран параметр *Показывать функцию агрегации*, элементы данных отображаются в скобках, перед которыми в легенде и подсказке указывается функция агрегации. Если для графика указан параметр *Метка набора данных*, метка отображается в скобках, перед которыми указывается функция агрегации.

Параметры отображения

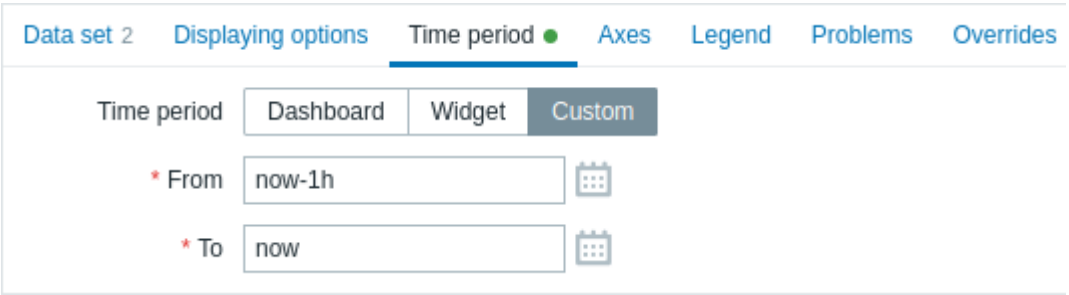
Вкладка **Параметры отображения** позволяет определить выбор данных истории:



Выбор данных истории	Укажите источник данных графика: **Авто** - данные выбираются в соответствии с классическим алгоритмом построения графика (по умолчанию); **История** - данные из истории; **Тренды** - данные из трендов.
Простые триггеры	Установите флажок, чтобы отображать пороговые значения простых триггеров. Пороговые значения отображаются пунктирными линиями с использованием цвета важности триггера. Простой триггер - это триггер с одной функцией (только `last`, `max`, `min`, `avg`) для одного элемента данных в выражении. Можно отобразить не более трех триггеров. Обратите внимание, что триггер должен находиться в отображаемом диапазоне, чтобы быть видимым.
Рабочее время	Установите флажок, чтобы отображать рабочее время на графике. Рабочее время (рабочие дни) отображается на графиках белым фоном, а нерабочее время - серым (при использовании темы веб-интерфейса *Original blue* по умолчанию).
Имена узлов сети в подписях	Выберите, отображать ли имена узлов сети в легенде графика и во всплывающей подсказке, которая появляется при наведении указателя на данные графика: **Авто** - имена узлов сети отображаются только в том случае, если в наборах данных присутствует более одного узла сети (по умолчанию). **Показать** - имена узлов сети отображаются. **Скрыть** - имена узлов сети скрыты. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.
Линия процентиля (слева)	Установите флажок и введите значение процентиля, чтобы отобразить указанный процентиль в виде линии на левой оси Y графика. Например, если задан процентиль 95%, линия процентиля будет находиться на уровне, ниже которого расположены 95 процентов значений.
Линия процентиля (справа)	Установите флажок и введите значение процентиля, чтобы отобразить указанный процентиль в виде линии на правой оси Y графика. Например, если задан процентиль 95%, линия процентиля будет находиться на уровне, ниже которого расположены 95 процентов значений.

Период времени

Вкладка **Период времени** позволяет задать период времени, за который будут отображаться данные на графике:

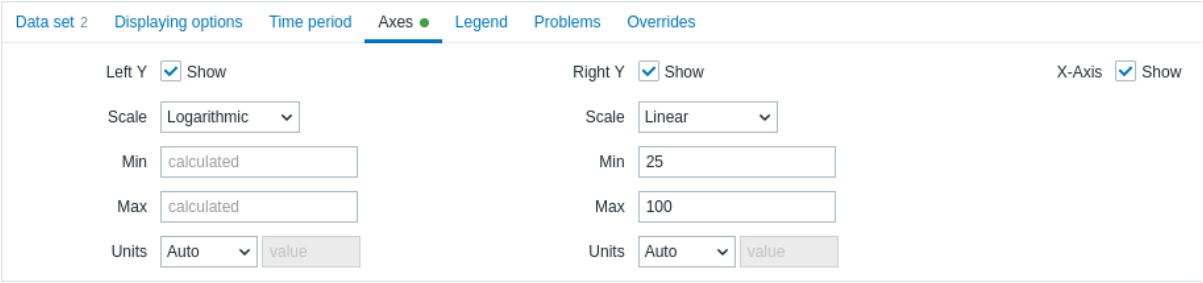


Период времени	Выберите источник данных для периода времени: **Панель мониторинга** - использовать селектор периода времени панели мониторинга; **Виджет** - использовать совместимый виджет (указанный в параметре *Виджет*); **Пользовательский** - использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении указателя мыши. Обратите внимание, что независимо от настройки *Период времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.

Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для параметра *Период времени* установлено значение «Виджет».
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».

Оси

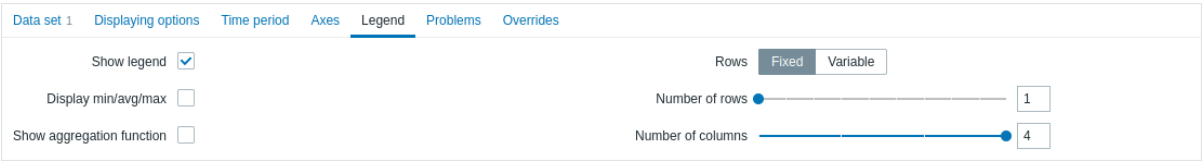
Вкладка **Оси** позволяет настроить отображение осей:



Левая ось Y	Установите этот флажок, чтобы сделать левую ось Y видимой. Флажок может быть недоступен, если он не выбран на вкладке *Набор данных* или *Переопределения*.
Правая ось Y	Установите этот флажок, чтобы сделать правую ось Y видимой. Флажок может быть недоступен, если он не выбран на вкладке *Набор данных* или *Переопределения*.
Ось X	Снимите этот флажок, чтобы скрыть ось X (по умолчанию флажок установлен).
Масштаб	Выберите масштаб значений оси графика в раскрывающемся списке: **Линейный** - значения оси увеличиваются на фиксированную величину (например, 10, 20, 30); подходит для данных, которые изменяются постепенно или охватывают малый либо средний диапазон; **Логарифмический** - значения оси увеличиваются экспоненциально (например, 10, 100, 1000); подходит для данных, которые быстро изменяются или охватывают большой диапазон.
Минимум	Задайте минимальное значение соответствующей оси. Указывается минимальное значение видимого диапазона оси Y.
Максимум	Задайте максимальное значение соответствующей оси. Указывается максимальное значение видимого диапазона оси Y.
Единицы измерения	Выберите единицу измерения значений оси графика в раскрывающемся списке: **Авто** - значения оси отображаются с использованием единицы измерения первого элемента данных в наборе данных; **Статические** - значения оси отображаются с использованием единицы измерения, указанной в поле ввода *значение*; если поле оставить пустым, отображаются только числовые значения.

Легенда

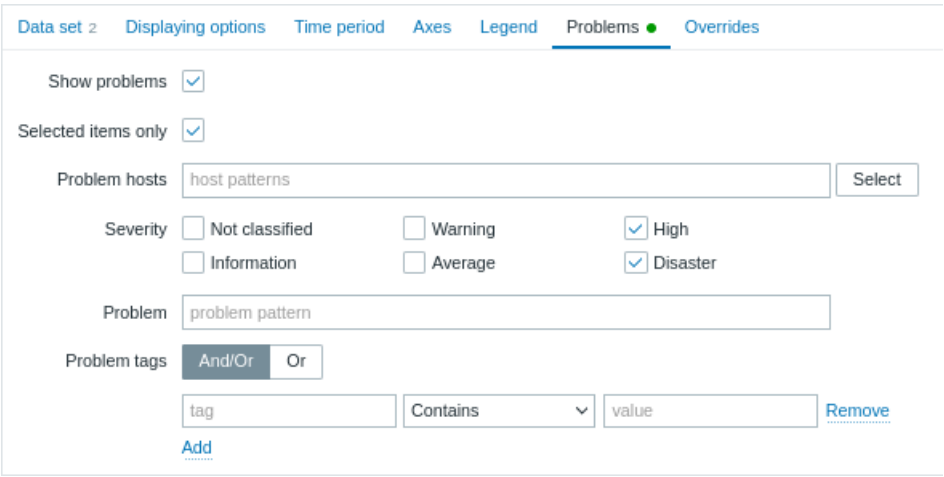
Вкладка **Легенда** позволяет настроить легенду графика:



Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на графике (по умолчанию флажок установлен).
Отображать мин./средн./макс.	Установите этот флажок, чтобы отображать в легенде минимальное, среднее и максимальное значения элемента данных.
Показывать функцию агрегации	Установите этот флажок, чтобы отображать в легенде функцию агрегации.
Строки	Выберите режим отображения строк легенды: **Фиксированное количество** - количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное количество** - количество отображаемых строк определяется количеством настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.
Количество строк/ Максимальное количество строк	Если для параметра *Строки* установлено значение «Фиксированное количество», укажите количество отображаемых строк легенды (1-10). Если для параметра *Строки* установлено значение «Переменное количество», укажите максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Укажите количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Отображать мин./средн./макс.* не установлен.

Проблемы

Вкладка **Проблемы** позволяет настроить отображение проблем:

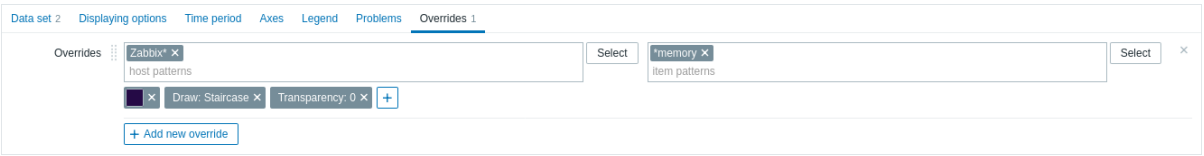


Показывать проблемы	Установите этот флажок, чтобы включить отображение проблем на графике (по умолчанию флажок не установлен, то есть отображение отключено).
Только выбранные элементы данных	Установите этот флажок, чтобы на графике отображались только проблемы выбранных элементов данных.
Узлы сети с проблемами	Выберите узлы сети с проблемами, которые должны отображаться на графике. Можно использовать шаблоны подстановки (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон подстановки, введите строку вручную и нажмите *Enter*. Во время ввода обратите внимание, что все соответствующие узлы сети отображаются в раскрывающемся списке. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.
Важность	Отметьте уровни важности проблем, чтобы отфильтровать проблемы, отображаемые на графике. Если уровни важности не отмечены, будут отображаться все проблемы.
Проблема	Укажите имя проблемы, которая должна отображаться на графике.

Теги проблем	Укажите теги проблем, чтобы ограничить количество проблем, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега будут объединены условием *ИЛИ*; **ИЛИ** - достаточно выполнения одного условия.

Переопределения

Вкладка **Переопределения** позволяет добавлять пользовательские переопределения для наборов данных:



Переопределения полезны, если для набора данных с помощью подстановочного символа * выбрано несколько элементов данных и необходимо изменить способ их отображения по умолчанию (например, базовый цвет или любое другое свойство).

Существующие переопределения (если они есть) отображаются в виде списка. Чтобы добавить новое переопределение:

- Нажмите кнопку 

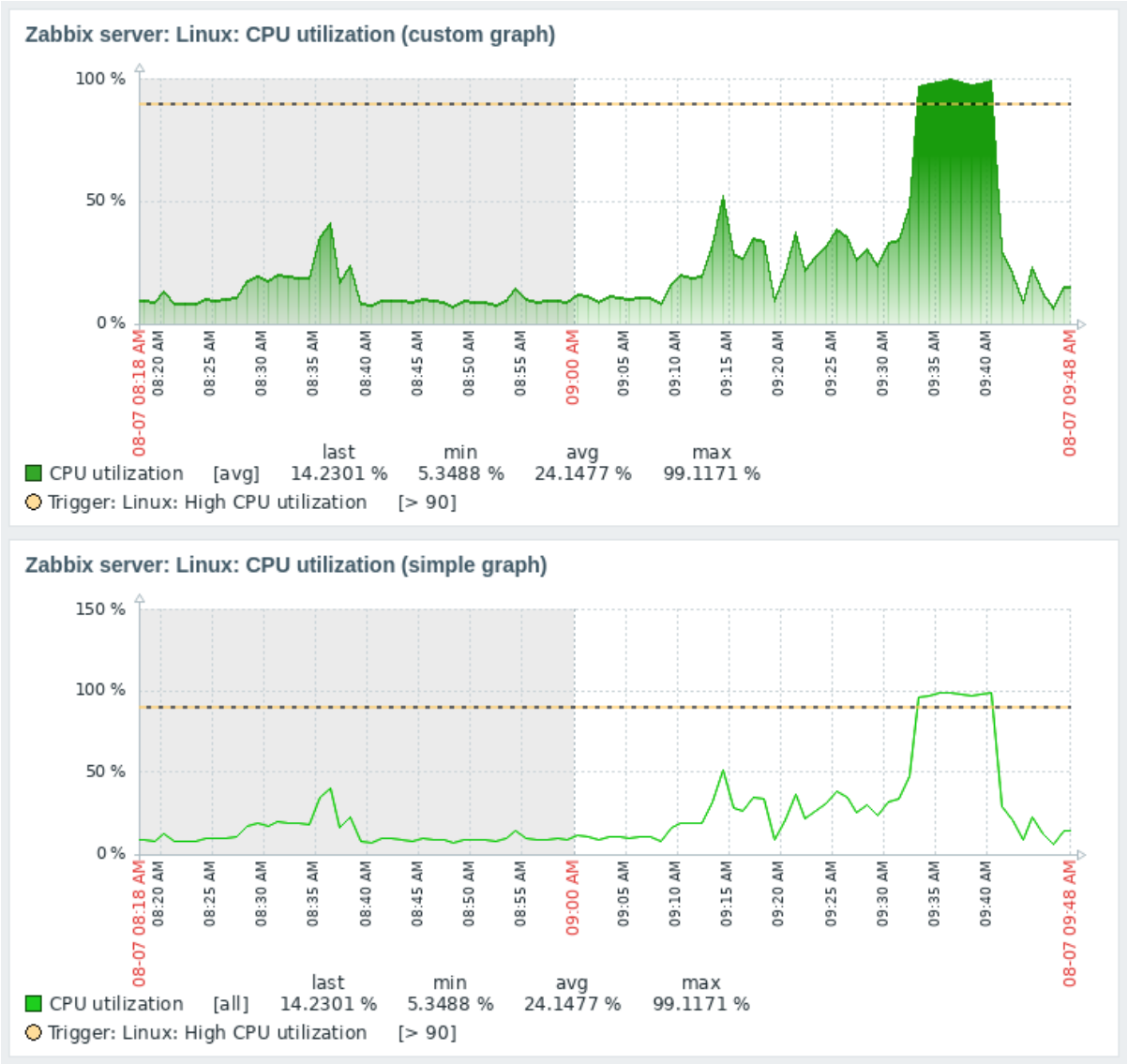
- Выберите узлы сети и элементы данных для переопределения. Также можно ввести шаблоны узлов сети и элементов данных. Можно использовать шаблоны с подстановочными символами (например, * возвращает результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным символом, введите строку вручную и нажмите *Enter*. Во время ввода обратите внимание, что все соответствующие узлы сети отображаются в раскрывающемся списке. Символ подстановки интерпретируется всегда, поэтому невозможно добавить, например, элемент данных с именем "item*" отдельно, если существуют другие соответствующие элементы данных (например, item2, item3). Параметры шаблона узла сети и шаблона элемента данных являются обязательными. Параметр для указания шаблонов узлов сети недоступен при настройке виджета на панели мониторинга шаблона. Параметр для указания списка элементов данных позволяет выбрать только элементы данных, настроенные в шаблоне при настройке виджета на панели мониторинга шаблона.
- Нажмите 

, чтобы выбрать параметры переопределения. Необходимо выбрать хотя бы один параметр переопределения. Описание параметров см. выше, на вкладке *Набор данных*.

График (классический)

Обзор

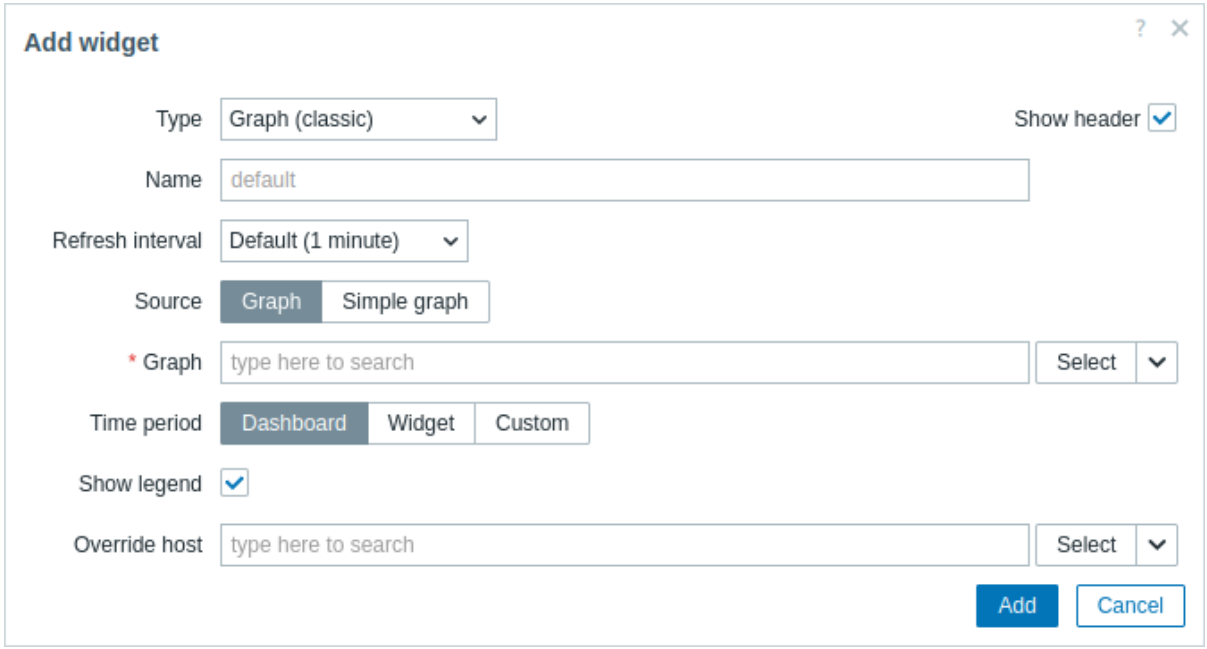
Виджет *График (классический)* отображает числовые данные элемента данных в виде изображения пользовательского графика или простого графика. Он поддерживает выбор периода времени, линии триггеров и необязательное отображение легенды.



Информацию, отображаемую в виджете *График (классический)*, можно скачать в виде изображения PNG, выбрав пункт *Скачать изображение* в меню *Действия* виджета ..., которое появляется при наведении указателя мыши на виджет. В зависимости от настроек браузера изображение автоматически сохраняется в папке `Downloads` либо открывается диалоговое окно *Сохранить*, позволяющее выбрать расположение файла.

Конфигурация

Чтобы настроить, выберите *Graph (classic)* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Source	Выберите тип графика: **Graph** - пользовательский график; **Simple graph** - простой график.
Graph	Выберите пользовательский график для отображения. Кроме того, можно выбрать совместимый виджет в качестве источника данных для графиков. Этот параметр доступен, если для *Source* задано значение "Graph".
Item	Выберите элемент данных для отображения на простом графике. Кроме того, можно выбрать совместимый виджет в качестве источника данных для элементов данных. Этот параметр доступен, если для *Source* задано значение "Simple graph".
Time period	Задайте временной период, за который нужно отображать данные на графике. Выберите источник данных для временного периода: **Dashboard** - использовать селектор временного периода панели; **Widget** - использовать совместимый виджет (задается в параметре *Widget*); **Custom** - использовать пользовательский временной период, заданный в параметрах *From* и *To*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении курсора. Обратите внимание, что независимо от настройки *Time period* виджета совместимые виджеты по-прежнему могут использовать его в качестве источника данных для временного периода.
Widget	Введите или выберите совместимый виджет в качестве источника данных для временного периода. Этот параметр доступен, если для *Time period* задано значение "Widget".
From	Введите или выберите начало временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w`, и т. д.). Этот параметр доступен, если для *Time period* задано значение "Custom".
To	Введите или выберите конец временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w`, и т. д.). Этот параметр доступен, если для *Time period* задано значение "Custom".
Show legend	Снимите этот флажок, чтобы скрыть легенду на графике (по умолчанию установлен).
Override host	Выберите совместимый виджет или селектор узла сети панели в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.

Легенда графика состоит из трех разделов:

- Элементы данных и их агрегированные значения
- Перцентили (если настроены)
- Триггеры (если какие-либо из них связаны с отображаемыми элементами данных)

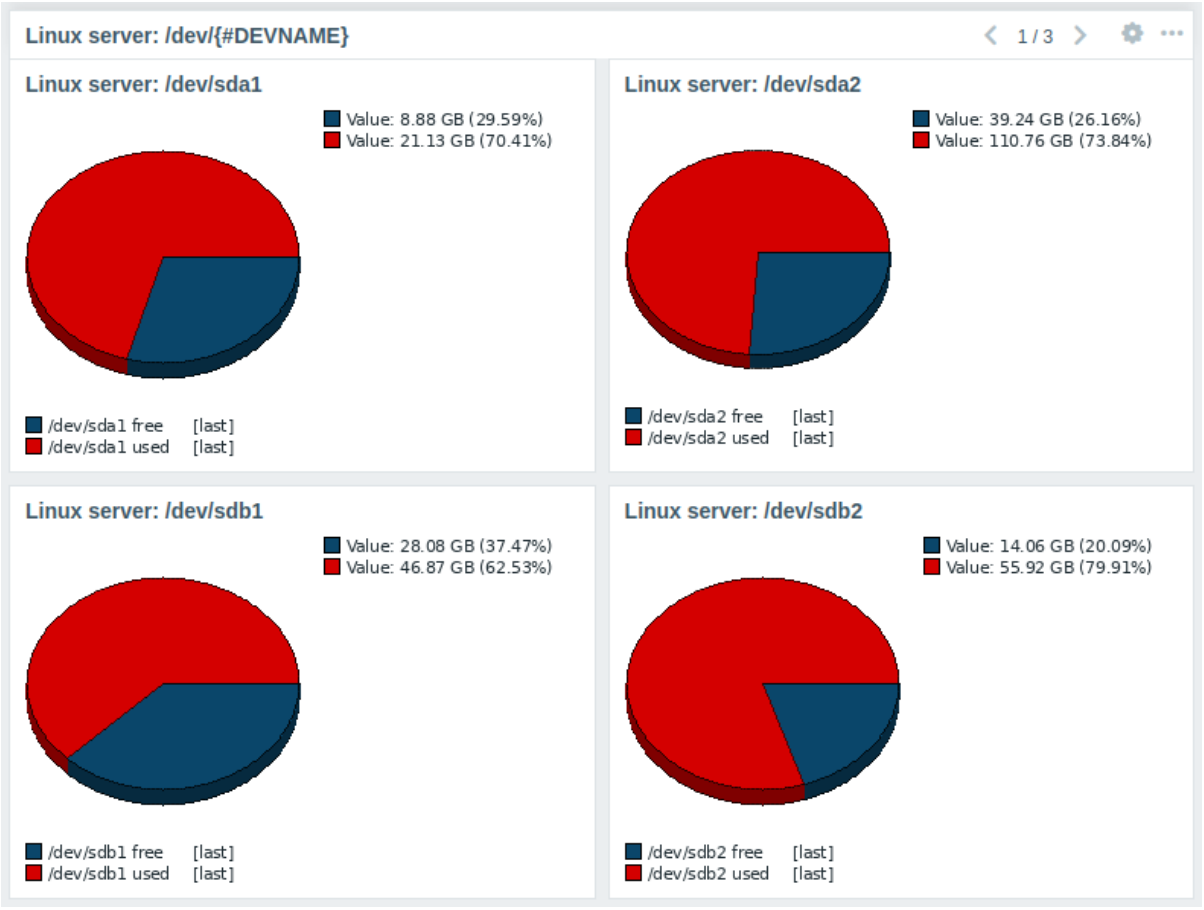
Если высоты графика внутри виджета недостаточно, легенда может не отображаться или отображаться только частично. Сначала скрываются триггеры и перцентили, затем легенда элементов данных. Чтобы отобразить полную легенду, увеличьте вертикальный размер виджета.

Note:
Можно отобразить не более 3 линий триггеров. Если триггеров больше, то для отображения приоритет получают триггеры с более низкой серьезностью.

Прототип графика

Обзор

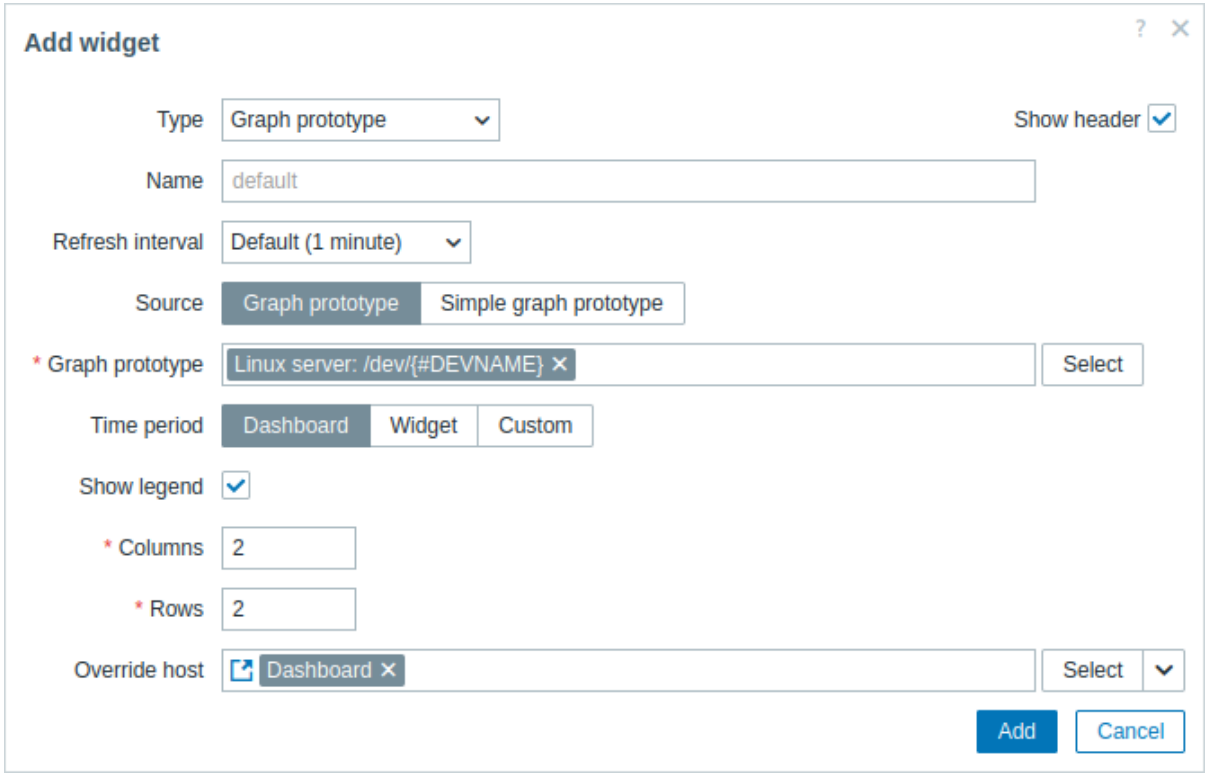
Виджет *Graph prototype* отображает сетку автоматически созданных пользовательских графиков на основе прототипов графиков или прототипов элементов данных правила обнаружения низкого уровня.



Графики располагаются в настраиваемой сетке; если количество графиков превышает размер сетки, в виджете включается постраничная навигация.

Настройка

Для настройки выберите *Прототип графика* в качестве типа виджета:



Помимо параметров, общих для всех виджетов, можно задать следующие специфические параметры:

Источник	Выберите источник графиков: **Прототип графика** или **Прототип простого графика**.
Прототип графика	Выберите прототип графика для отображения графиков, обнаруженных прототипом графика. Этот параметр доступен, если для параметра *Источник* выбрано значение «Прототип графика».
Прототип элемента данных	Выберите прототип элемента данных для отображения простых графиков для элементов данных, обнаруженных прототипом элемента данных. Этот параметр доступен, если для параметра *Источник* выбрано значение «Прототип простого графика».
Период времени	Задайте период времени, за который будут отображаться данные на графиках. Выберите источник данных для периода времени: **Панель мониторинга** — использовать селектор периода времени панели мониторинга; **Виджет** — использовать совместимый виджет (заданный в параметре *Виджет*); **Пользовательский** — использовать пользовательский период времени, заданный в параметрах *От* и *До*; если задано это значение, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении указателя мыши. Обратите внимание, что независимо от настройки *Период времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для параметра *Период времени* выбрано значение «Виджет».
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* выбрано значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* выбрано значение «Пользовательский».
Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на графиках (по умолчанию флажок установлен).
Переопределить узел сети	Выберите совместимый виджет или селектор узла сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.

Столбцы	Введите количество столбцов графиков, отображаемых в виджете прототипа графика.
Строки	Введите количество строк графиков, отображаемых в виджете прототипа графика.

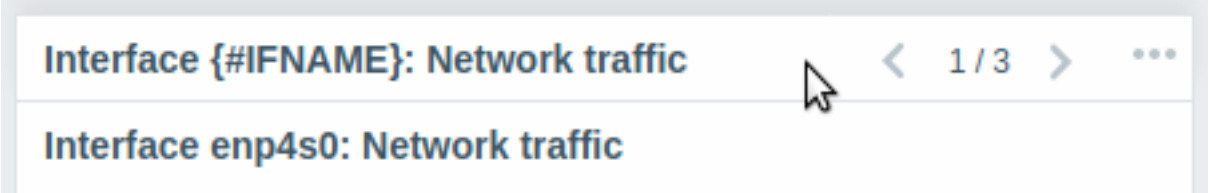
Легенда графика состоит из трех разделов:

- Элементы данных и их агрегированные значения
- Процентили (если настроены)
- Триггеры (если связаны с отображаемыми элементами данных)

Если высота графика в виджете недостаточна, легенда может не отображаться или отображаться только частично. Сначала скрываются триггеры и процентили, затем легенда элементов данных. Чтобы отобразить легенду полностью, увеличьте вертикальный размер виджета.

Note:
Можно отображать не более 3 линий триггеров. Если триггеров больше, при отображении приоритет отдается триггерам с более низким уровнем серьезности.

Параметры *Столбцы* и *Строки* позволяют разместить в виджете несколько графиков, однако обнаруженных графиков все равно может быть больше, чем столбцов или строк в виджете. В этом случае в виджете становится доступна постраничная навигация, а в выдвижном заголовке можно переключаться между страницами с помощью стрелок влево и вправо:

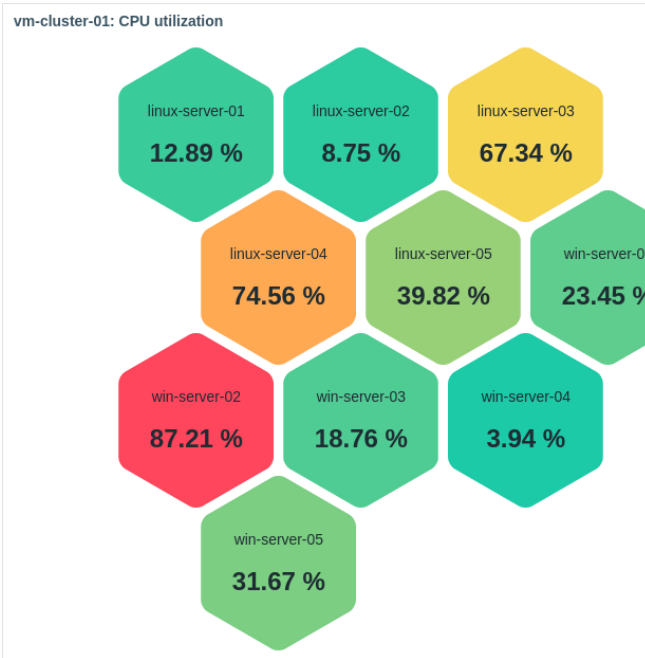


Honeycomb

Обзор

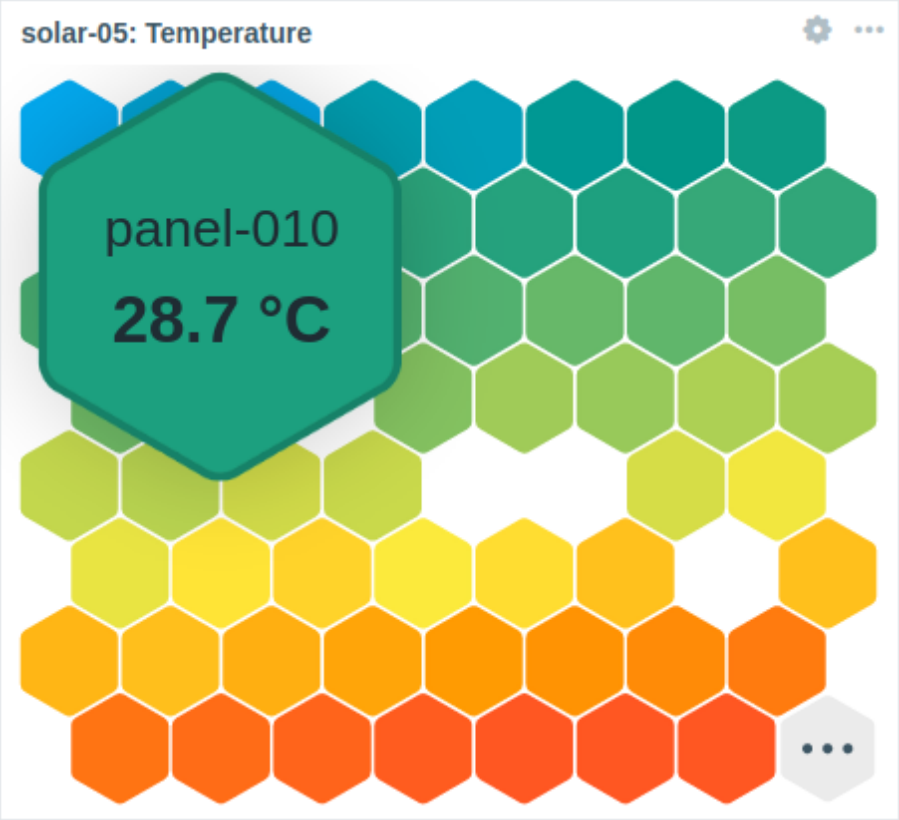
Виджет *Honeycomb* отображает узлы сети или элементы данных в виде шестиугольных ячеек в сетке. Такое представление упрощает обнаружение проблемных узлов сети и элементов данных, выявление кластеров проблем, сравнение групп и многое другое.





Количество отображаемых ячеек ограничено размером виджета и минимальным размером ячейки (32px). Если все ячейки не помещаются, в последней ячейке отображается многоточие. Размер виджета можно изменять, при этом количество, размер и расположение ячеек корректируются динамически.

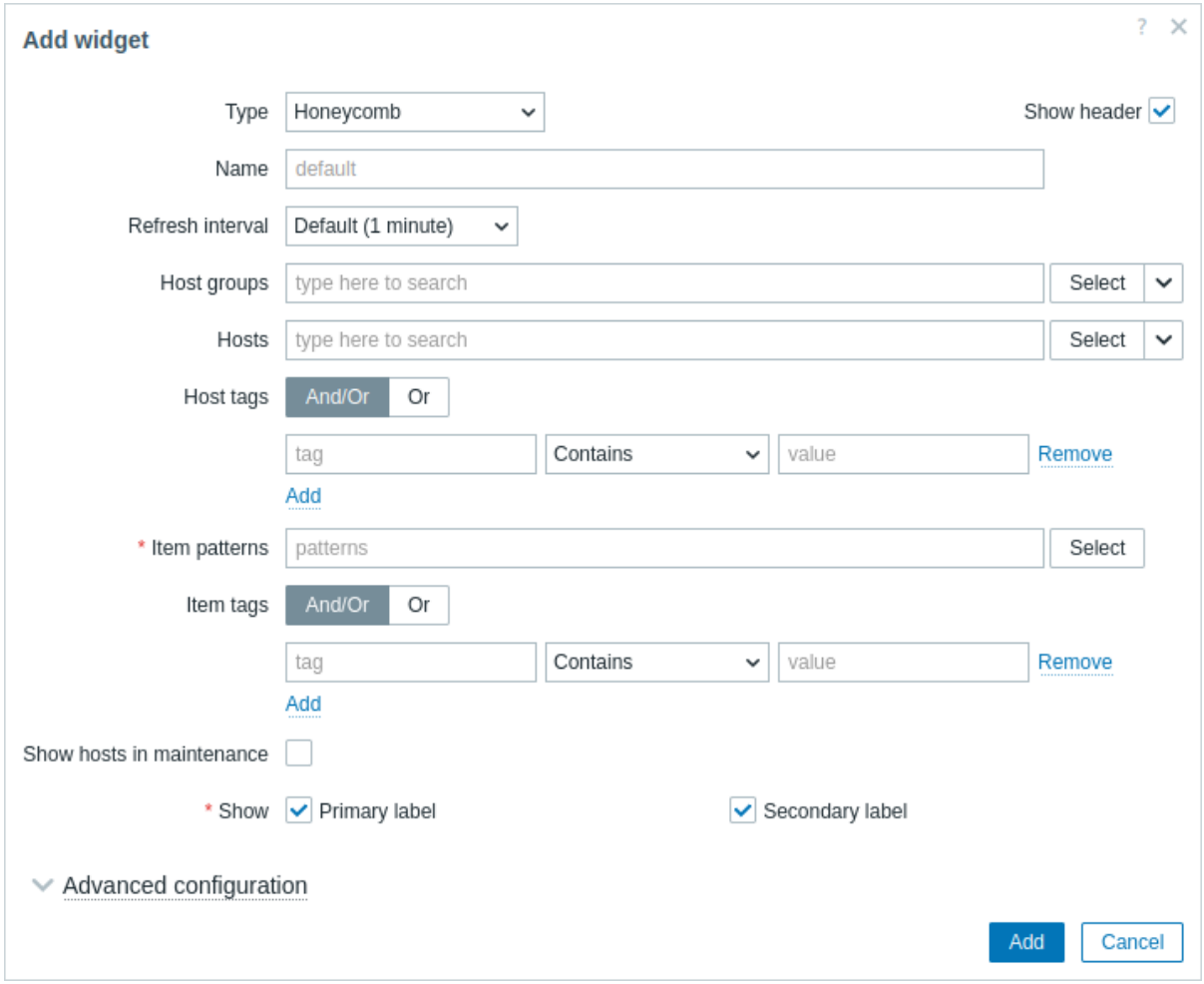
При наведении указателя мыши выбранная ячейка увеличивается для улучшения видимости. При нажатии на ячейку ее граница выделяется до тех пор, пока не будет выбрана другая ячейка.



Информацию, отображаемую в виджете *Honeycomb*, можно скачать в виде изображения PNG, выбрав пункт *Download image* в меню *Actions* виджета ⋯, которое появляется при наведении указателя мыши на виджет. В зависимости от настроек браузера изображение либо автоматически сохраняется в папке `Downloads`, либо открывается диалоговое окно *Save*, позволяющее выбрать расположение файла.

Конфигурация

Чтобы настроить, выберите *Honeycomb* в качестве типа:



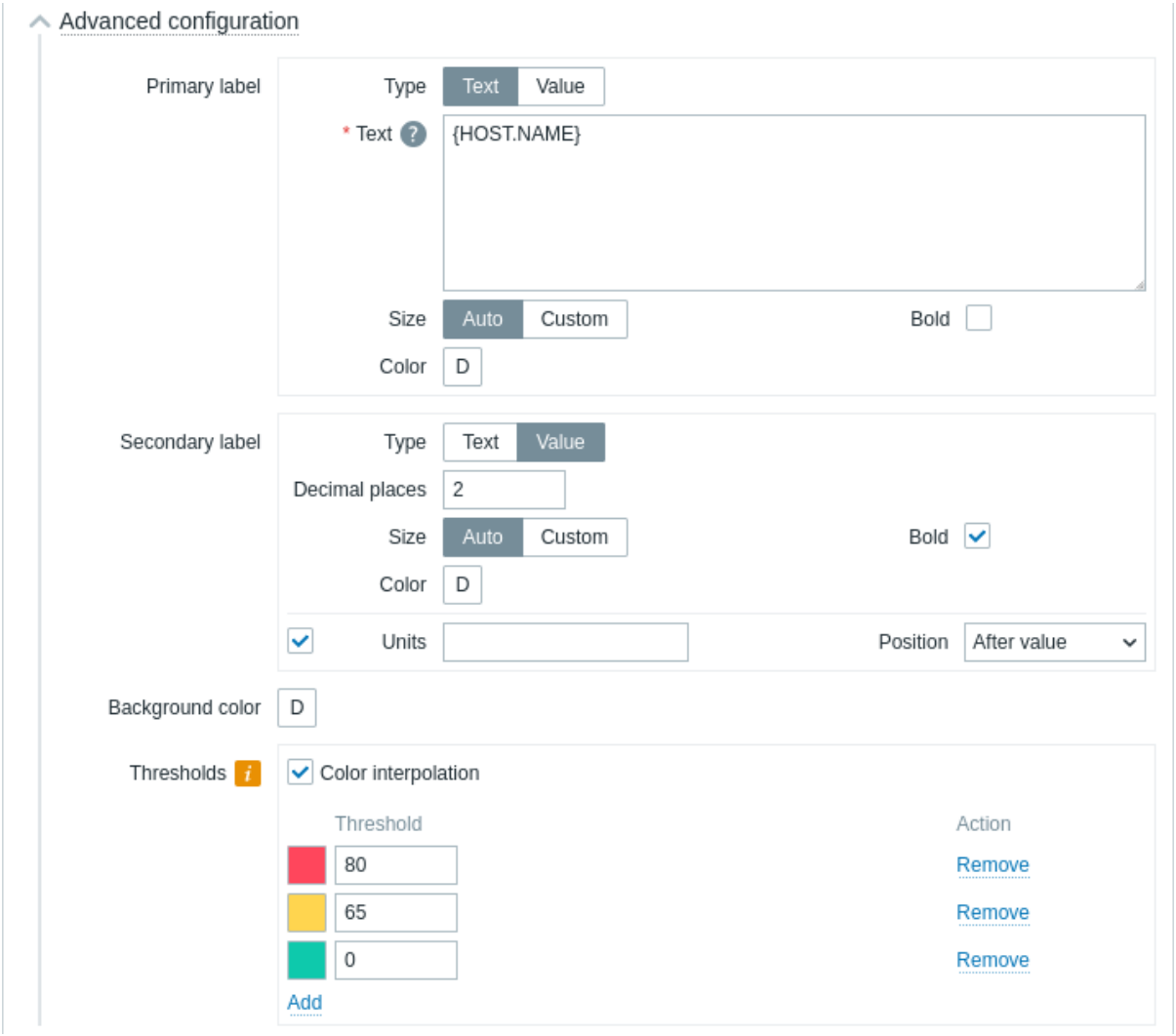
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети. Либо выберите совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет показан выпадающий список подходящих групп. При выборе родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети; если группы узлов сети не выбраны, виджет отобразит все группы узлов сети, содержащие узлы сети с элементами данных, соответствующими выбранному *Шаблону элемента данных* (см. ниже). Этот параметр недоступен при настройке виджета на шаблонной панели.
Узлы сети	Выберите узлы сети. Либо выберите совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет показан выпадающий список подходящих узлов сети. Если узлы сети не выбраны, виджет отобразит все узлы сети с элементами данных, соответствующими выбранному *Шаблону элемента данных* (см. ниже). Этот параметр недоступен при настройке виджета на шаблонной панели.

Теги узлов сети	Укажите теги, чтобы ограничить количество узлов сети, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно выполнения одного условия. Этот параметр недоступен при настройке виджета на шаблонной панели.
Шаблоны элементов данных	Введите шаблоны элементов данных или выберите существующие элементы данных в качестве шаблонов элементов данных. На honeycomb будут отображаться данные элементов данных, соответствующих введенным или выбранным шаблонам. Параметр *Шаблоны элементов данных* обязателен. Для выбора можно использовать шаблоны с подстановочными знаками (например, `*` вернет элементы данных, соответствующие нулю или более символам; `Zabbix*` вернет элементы данных, начинающиеся с ”Zabbix”). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. При вводе будет показан выпадающий список подходящих элементов данных, ограниченный элементами, относящимися к выбранным *Узлам сети* или узлам сети в выбранных *Группах узлов сети*, если они указаны. Символ подстановки всегда интерпретируется, поэтому невозможно, например, добавить элемент данных с именем *item** отдельно, если есть другие подходящие элементы данных (например, item2, item3). При настройке виджета на шаблонной панели этот параметр позволяет выбирать только элементы данных, настроенные на шаблоне.
Теги элементов данных	Укажите теги, чтобы ограничить количество элементов данных, отображаемых в виджете. Дополнительные сведения см. выше в разделе *Теги узлов сети*.
Показывать узлы сети в обслуживании	Установите этот флажок, чтобы отображать узлы сети в обслуживании (в этом случае рядом с именем узла сети будет показан значок обслуживания). При настройке виджета на шаблонной панели этот параметр называется *Показывать данные в обслуживании*.
Показывать	Установите этот флажок, чтобы отображать соответствующий элемент ячейки honeycomb - основную метку, вторичную метку. Должен быть выбран как минимум один элемент.
Дополнительная настройка	Нажмите заголовок *Дополнительная настройка*, чтобы отобразить параметры дополнительной настройки.

Расширенная настройка

Параметры расширенной настройки доступны а сворачиваемом разделе *Расширенная настройка (Advanced configuration)* и только для элементов, отмеченных в поле *Показать (Show)* (см. выше), а также для цвета фона или пороговых значений для ячеек сот.



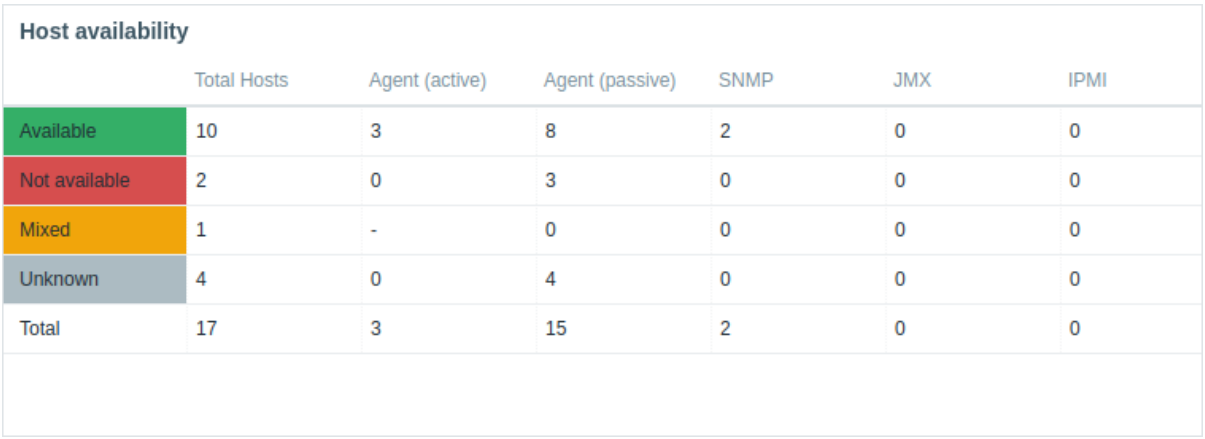
Первичная/Вторичная метка (Primary/Secondary label)	
Тип (Type)	Выберите тип метки: **Текст (Text)** — метка будет отображать текст, указанный для параметра *Текст (Text)*; **Значение (Value)** — метка будет отображать значение элемента данных с количеством десятичных знаков, как указано для параметра *Число десятичных знаков (Decimal places)*.
Текст (Text)	Введите текст метки. Этот текст может переопределять имя элемента данных по умолчанию. Поддерживается многострочный текст. Возможна комбинация текста и поддерживаемых макросов. Поддерживаются `{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}`, а также пользовательские макросы. Ячейки сот сортируются в алфавитном порядке по имени узла сети и затем, для каждого узла сети — по имени элемента данных. Этот параметр доступен, если *Тип (Type)* имеет значение *Текст (Text)*.
Число десятичных знаков (Decimal places)	Введите количество десятичных знаков для отображения значения. Этот параметр доступен, если *Тип (Type)* имеет значение *Значение (Value)*, и имеет смысл только для элементов данных, которые возвращают числовые (с плавающей точкой) значения.
Размер (Size)	Выберите размер метки: **Автоматически (Auto)** — использовать размер метки, настраиваемый автоматически; **Пользовательское (Custom)** — ввести пользовательский размер метки (в процентах, относительно размера ячейки сот). Обратите внимание, что метки, которые не вмещаются в размер ячейки сот, усекаются.
Жирный (Bold)	Установите флажок для отображения подписи данных жирным шрифтом.

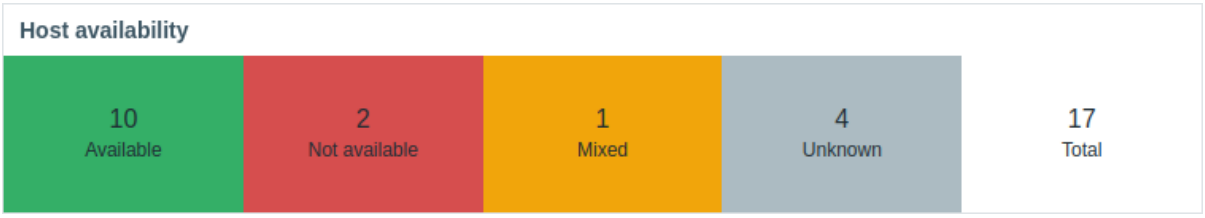
Цвет (Color)	Выберите цвет подписи из палитры. «D» означает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуться к цвету по умолчанию, нажмите в палитре на кнопку *Использовать умолчание (Use default)*.
Единицы измерения (Units)	
Единицы измерения (Units)	Установите флажок для отображения единиц измерения со значением элемента данных. Если вы вводите название единиц измерения, оно будет переопределять единицы измерения, заданные в настройках элемента данных. Этот параметр доступен, если *Тип (Type)* имеет значение «Значение (Value)».
Расположение (Position)	Выберите расположение единиц измерения элемента данных (до или после значения элемента данных). Этот параметр игнорируется для следующих единиц измерения, относящихся ко времени: unixtime, uptime, s. Этот параметр доступен, если *Тип (Type)* имеет значение «Значение (Value)», а параметр *Единицы измерения (Units)* отмечен.
Цвет фона (Background color)	
Цвет фона (Background color)	Выберите из палитры цвет фона ячейки сот. «D» означает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуться к цвету по умолчанию, нажмите в палитре на кнопку *Использовать умолчание (Use default)*.
Пороги (Thresholds)	
Интерполяция цветов (Color interpolation)	Установите флажок, чтобы разрешить для ячеек сот плавный переход между цветами пороговых значений. Этот параметр доступен, если заданы два или более пороговых значения.
Порог (Threshold)	Нажмите *Добавить (Add)* для добавления порога, выберите из палитры цвет порога и укажите числовое значение. Список порогов будет при сохранении отсортирован в возрастающем порядке. Обратите внимание, что цвета, настроенные для порогов, будут правильно отображаться только для числовых элементов данных. Поддерживаются суффиксы (например, «1d», «2w», «4K», «8G»). Поддерживаются преобразования значений.

Доступность узла сети

Обзор

Виджет *Доступность узлов сети* отображает доступность узлов сети в выбранных группах узлов сети, позволяя отслеживать количество узлов сети, которые доступны или недоступны.



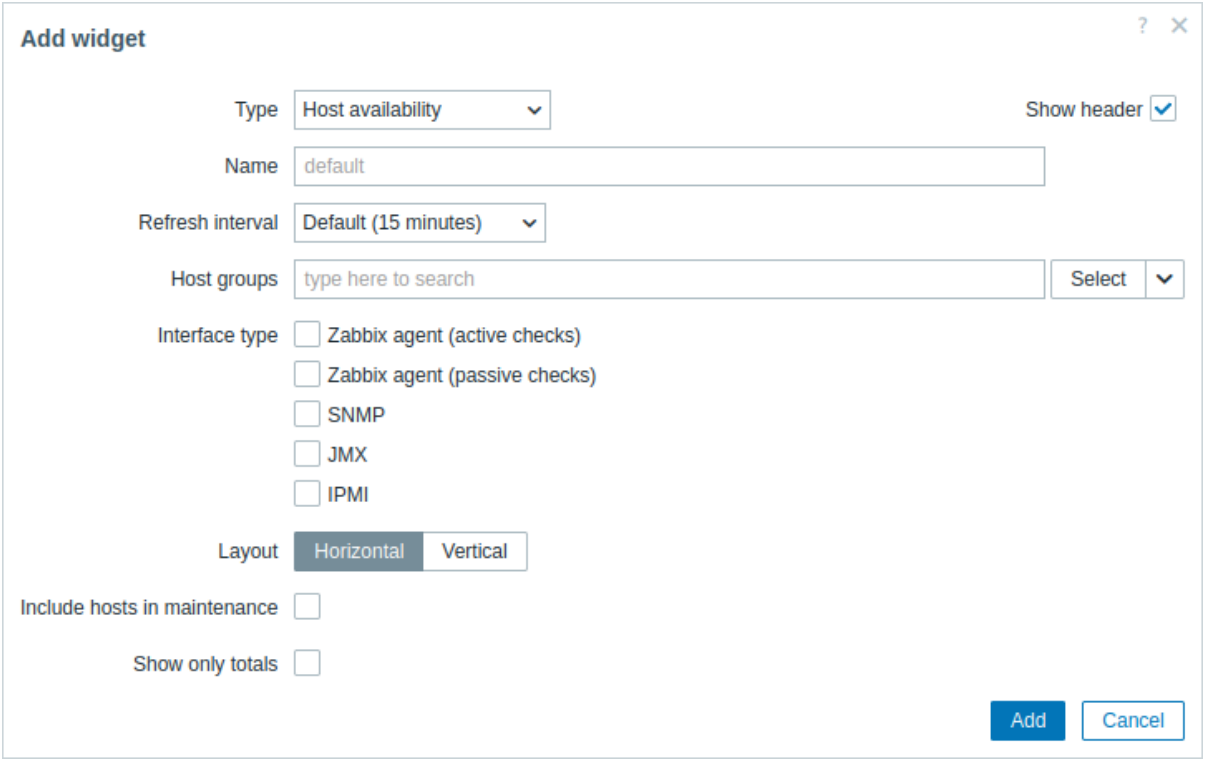


Доступность узлов сети рассчитывается следующим образом:

- *Доступны* - узлы сети, у которых доступны все интерфейсы
- *Недоступны* - узлы сети, у которых все интерфейсы недоступны
- *Смешанные* - узлы сети, у которых как минимум один интерфейс недоступен и как минимум один доступен или неизвестен
- *Неизвестно* - узлы сети, у которых как минимум один интерфейс имеет статус неизвестен, но ни один не недоступен
- *Всего* - общее количество всех узлов сети

Конфигурация

Чтобы настроить, выберите *Host availability* в качестве типа:



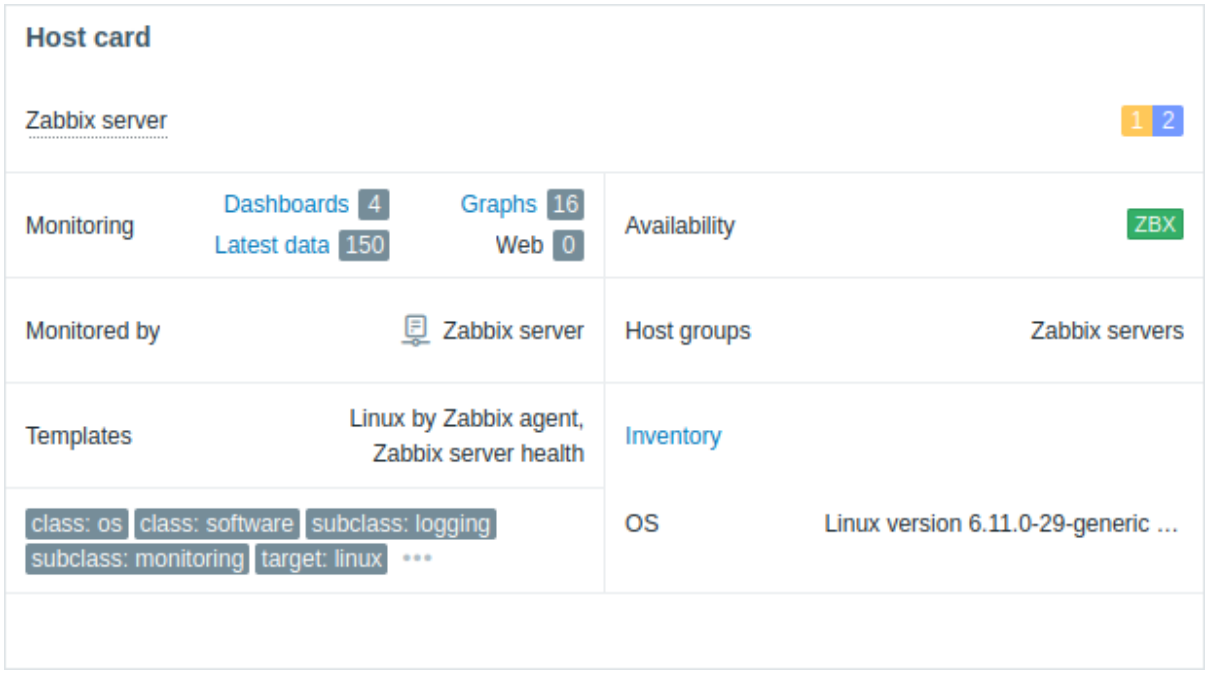
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Host groups	Выберите группы узлов сети. Либо выберите совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на шаблонной панели.
Interface type	Выберите, для каких интерфейсов узла сети нужно отображать данные доступности — *Zabbix agent (active checks)*, *Zabbix agent (passive checks)*, *SNMP*, *JMX*, *IPMI*. Для *Zabbix agent (active checks)* ячейка *Mixed* всегда будет пустой, поскольку этот тип элементов данных не может иметь несколько интерфейсов. Если ничего не выбрано, по умолчанию отображается доступность всех интерфейсов.
Layout	Выберите горизонтальное отображение (столбцы) или вертикальное отображение (строки).
Include hosts in maintenance	Включать в статистику узлы сети, находящиеся на обслуживании. При настройке виджета на шаблонной панели этот параметр называется *Show data in maintenance*.
Show only totals	Если флажок установлен, отображается только итог по узлам сети без разбивки по интерфейсам. Этот параметр недоступен, если выбран только один интерфейс.

Карточка узла сети

Обзор

Виджет *Карточка узла сети* отображает сведения об одном узле сети, позволяя быстро оценить состояние и конфигурацию узла сети (доступность, проблемы, инвентарные данные и т. д.).

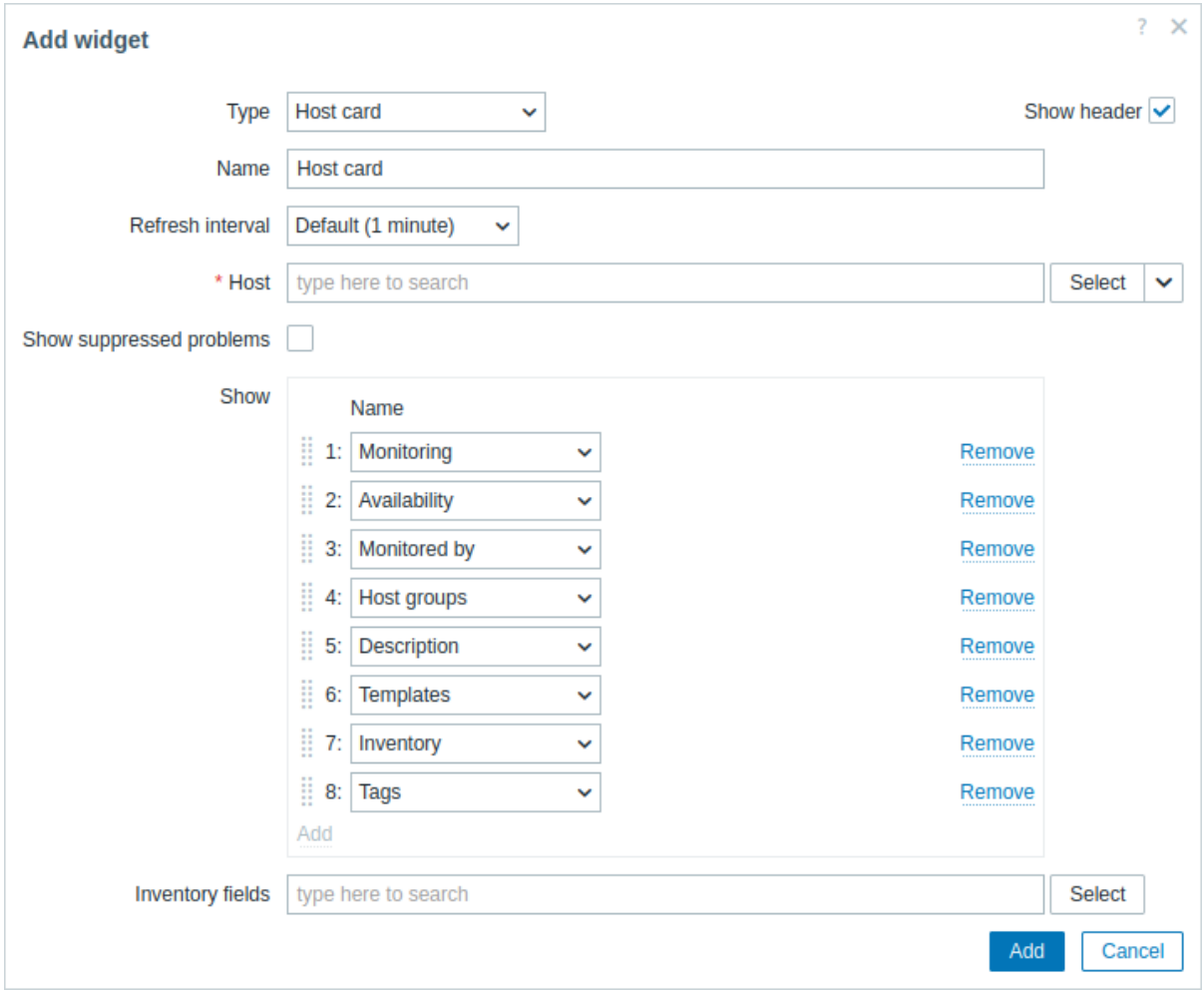


Узел сети можно задать непосредственно в настройках виджета или выбрать либо из совместимого виджета, либо с помощью селектора узла сети на панели мониторинга.

Виджет может отображать несколько разделов, каждый из которых показывает различную информацию. Макет разделов автоматически подстраивается в зависимости от ширины виджета; при расширении по горизонтали разделы автоматически перераспределяются в несколько столбцов.

Настройка

Для настройки выберите тип *Карточка узла сети*:



В дополнение к параметрам, общим для всех виджетов, вы можете задать следующие специальные параметры:

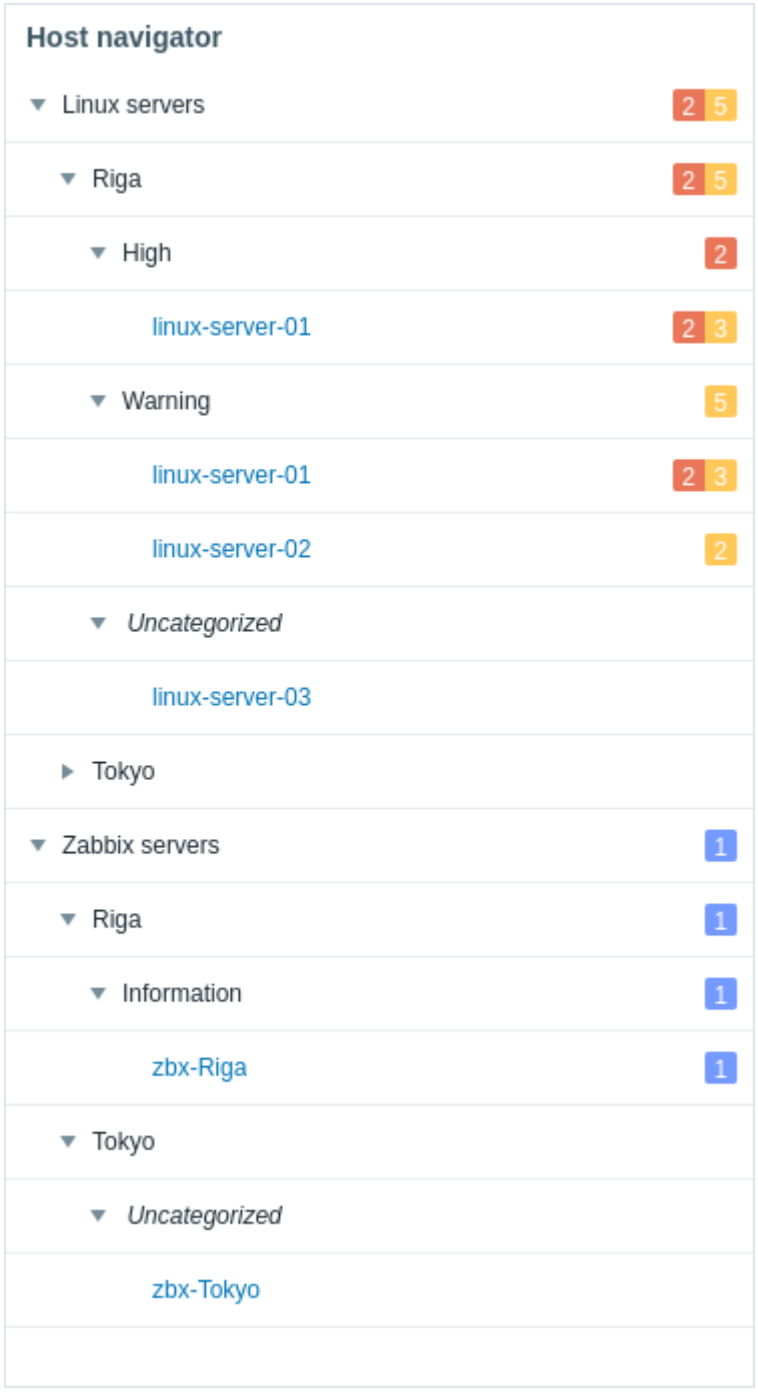
Узел сети	Выберите узел сети. Либо выберите совместимый виджет или панель как источник данных для узлов сети. По мере ввода будут появляться подходящие варианты. Верхняя часть виджета всегда отображает **имя узла сети** (нажатие на него открывает меню узла сети): - Если узел сети отключен, рядом с именем узла сети отображается красная метка *Отключен*. - Если узел сети находится на обслуживании, рядом с именем узла сети отображается значок обслуживания. Верхняя часть виджета также отображает количество **проблем узла сети**, сгруппированных по важности (нажатие на них открывает проблемы узла сети). Этот параметр недоступен при настройке виджета на панели шаблона.
Показывать подавленные проблемы	Установите этот флажок, чтобы включить подавленные проблемы в расчет проблем.
Показывать	Добавьте разделы для отображения в виджете. Разделы можно переставлять с помощью перетаскивания.
Группы узлов сети	Отображает группы узлов сети, в которые входит узел сети.
Описание	Отображает описание узла сети.
Мониторинг	Отображает быстрые ссылки для перехода к Панелям, Последним данным, Графикам и Веб-сценариям узла сети. Число рядом с каждой ссылкой указывает количество соответствующих сущностей.
Доступность	Отображает доступность узла сети по интерфейсам.
Кем мониторится	Отображает, мониторится ли узел сети сервером Zabbix, определенным прокси или группой прокси.
Шаблоны	Отображает шаблоны, связанные с узлом сети.

Инвентарные данные	Отображает поля инвентарных данных узла сети.
Теги	Отображает теги узла сети.
Поля инвентарных данных	Выберите поля инвентарных данных для отображения. По мере ввода будут появляться подходящие варианты. Если поля инвентарных данных не указаны, будут отображены все заполненные поля инвентарных данных. Этот параметр доступен, если *Показывать* содержит *Инвентарные данные*.

Навигатор узла сети

Обзор

Виджет *Host navigator* отображает список узлов сети на основе различных параметров фильтрации и группировки.



Группы, по которым организованы узлы сети, можно разворачивать и сворачивать.

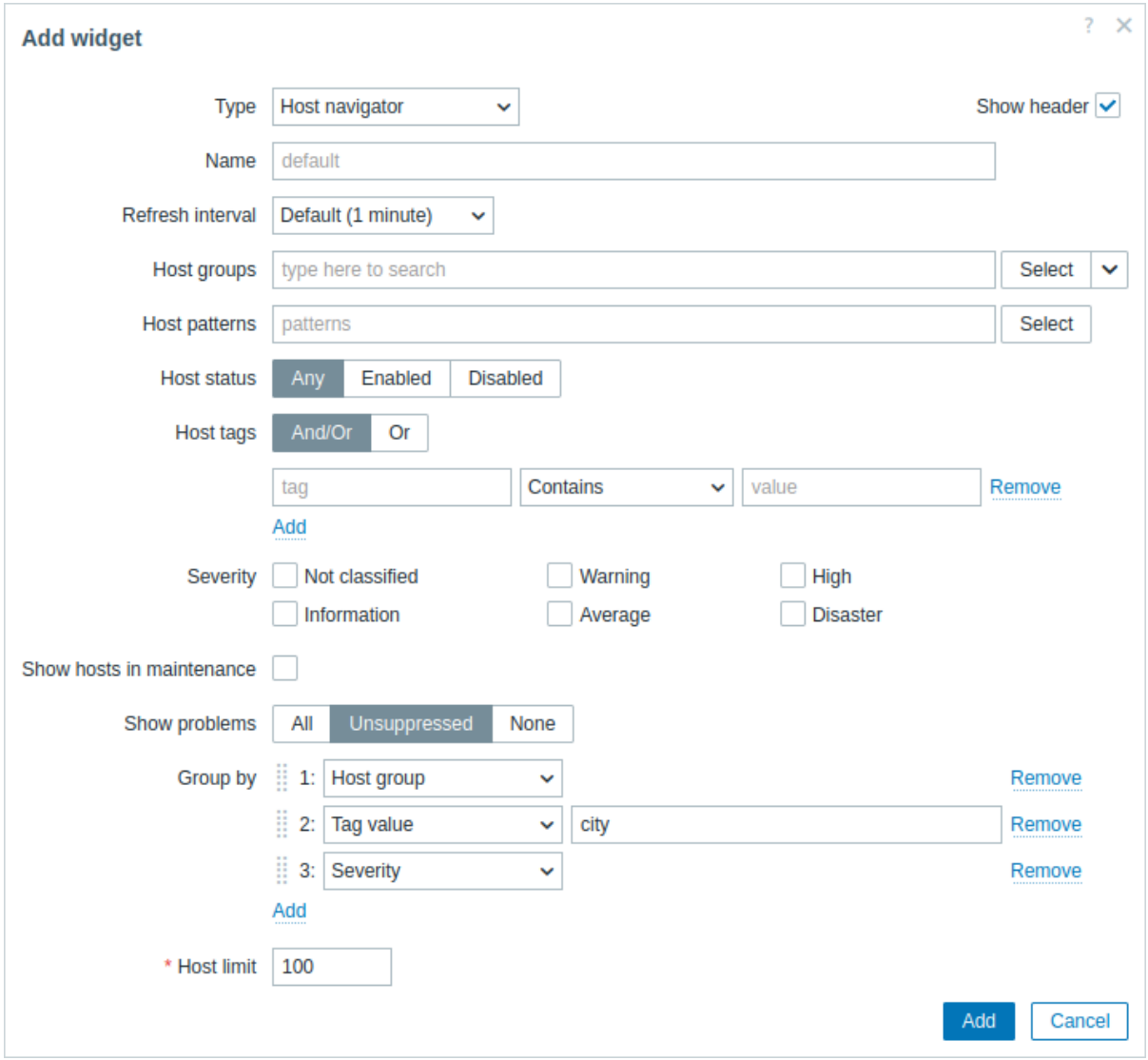
Для групп, проблем и узлов сети в обслуживании доступны дополнительные сведения во всплывающих подсказках при наведении мыши.

Этот виджет наиболее полезен для управления тем, что отображают другие виджеты, в зависимости от выбранного узла сети.



Настройка

Для настройки выберите тип *Навигатор узлов сети*:



Помимо параметров, общих для всех виджетов, можно задать следующие специфические параметры:

Группы узлов сети	Выберите группы узлов сети. Также можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение: при вводе имени группы появится раскрывающийся список соответствующих групп. Выбор родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети; если группы узлов сети не выбраны, виджет отобразит все узлы сети из всех групп узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Шаблоны узлов сети	Введите шаблоны узлов сети или выберите существующие узлы сети в качестве шаблонов. Узлы сети, соответствующие указанным шаблонам, будут отображаться в навигаторе узлов сети. Это поле поддерживает автодополнение: при вводе имени узла сети появится раскрывающийся список соответствующих узлов сети. Если узлы сети не выбраны, виджет отобразит все узлы сети. Для выбора можно использовать шаблоны с подстановочными знаками (например, `*` вернет узлы сети, соответствующие нулю или более символам; `Zabbix*` вернет узлы сети, начинающиеся с «Zabbix»). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. При начале ввода в раскрывающемся списке будут показаны соответствующие узлы сети, ограниченные узлами сети, входящими в выбранные *Группы узлов сети*, если таковые выбраны. Этот параметр недоступен при настройке виджета на панели шаблона.

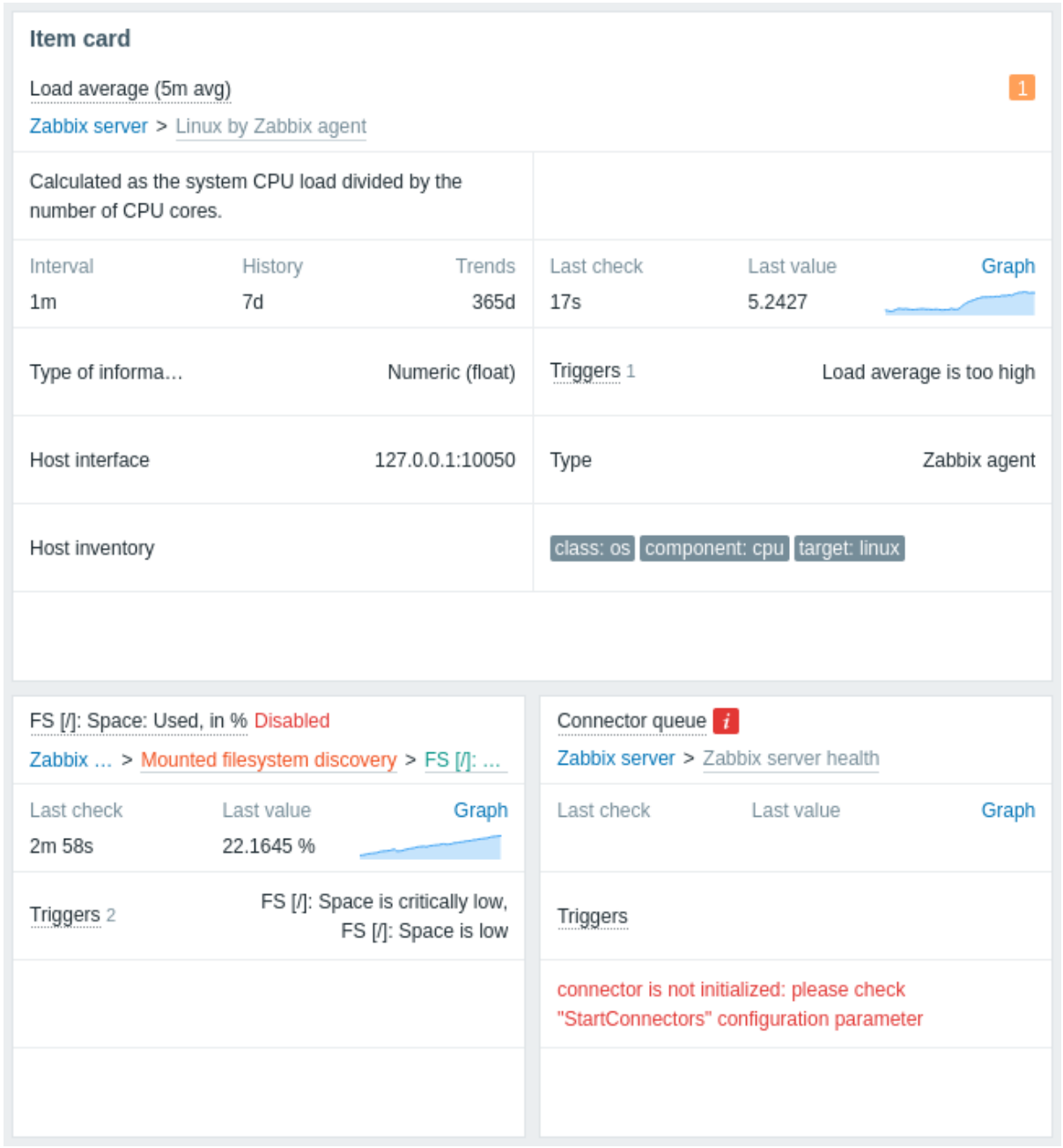
Состояние узлов сети	Фильтруйте отображаемые узлы сети по их состоянию (любое, включены, отключены). Этот параметр недоступен при настройке виджета на панели шаблона.
Теги узлов сети	Укажите теги для фильтрации узлов сети, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (сопоставление подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (сопоставление подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега будут объединены оператором ИЛИ; **ИЛИ** - достаточно выполнения одного условия.
Важность	Этот параметр недоступен при настройке виджета на панели шаблона. Отметьте важности проблем для фильтрации узлов сети с проблемами, отображаемых в виджете. Если важности не отмечены, будут отображаться все узлы сети со всеми проблемами.
Показывать узлы сети в режиме обслуживания	Установите этот флажок, чтобы отображать узлы сети в режиме обслуживания (в этом случае рядом с именем узла сети будет показан значок обслуживания). При настройке виджета на панели шаблона этот параметр называется *Показывать данные в режиме обслуживания*.
Показывать проблемы	Фильтруйте отображаемые проблемы по их состоянию (все, неподтвержденные, нет). Проблемы обозначаются цветными блоками в конце строки узла сети. Цвета зависят от важности проблемы, которую можно настроить в разделе *Администрирование* > *Общие* > *Параметры отображения триггеров*. Важность проблем можно изменить при обновлении проблем.

Группировать по	Добавьте атрибут группировки, по которому будут сгруппированы выбранные узлы сети: **Группа узлов сети** - группировать узлы сети по их группе; **Значение тега** - введите имя тега, чтобы сгруппировать узлы сети по значениям этого тега (например, введите «city», чтобы сгруппировать узлы сети по значениям «Riga», «Tokyo» и т. д.); **Важность** - группировать узлы сети по важности их проблем. Если параметр *Показывать проблемы* настроен на отображение проблем, они отображаются следующим образом: - для каждой группы важности отображается только соответствующее количество проблем; - для каждого узла сети отображается количество всех его проблем. Обратите внимание, что узлы сети будут сгруппированы только по важностям, отмеченным в параметре *Важность*; если важности не отмечены, все узлы сети будут сгруппированы по всем важностям. Атрибуты группировки можно менять местами, перетаскивая их вверх или вниз за маркер перед именем группы. Обратите внимание, что порядок атрибутов группировки определяет порядок вложенности групп. Например, указание нескольких имен тегов (1: цвет, 2: город) приведет к группировке узлов сети сначала по цвету (красный, синий и т. д.), а затем по городу (Riga, Tokyo и т. д.). Узел сети может отображаться в нескольких группах в зависимости от настроенных атрибутов группировки (например, при группировке по группе узлов сети, если узел сети входит в несколько групп узлов сети). При нажатии на такой узел сети он выбирается и выделяется во всех группах. Узлы сети, не соответствующие настроенным атрибутам группировки, отображаются в группе *Без категории*. Можно указать до 10 атрибутов группировки, и все они должны быть уникальными. Если атрибуты группировки не указаны, узлы сети не будут сгруппированы.
Ограничение количества узлов сети	Введите максимальное количество отображаемых узлов сети. Допустимые значения находятся в диапазоне от 1 до 9999. Если доступно больше узлов сети, чем задано ограничением, под отображаемыми узлами сети появится соответствующее сообщение (например, «Показано 100 из более чем 100 узлов сети»). Обратите внимание, что заданное ограничение количества узлов сети также влияет на отображение настроенных групп; например, если ограничение установлено равным 100, а узлы сети сгруппированы по значениям тегов (более 200), в виджете будут отображаться только первые 100 значений тегов с соответствующими узлами сети. На этот параметр не влияет параметр *Ограничение для результатов поиска и фильтрации* в разделе *Администрирование* > *Общие* > *Веб-интерфейс*. Этот параметр недоступен при настройке виджета на панели шаблона.

Карточка элемента данных

Обзор

Виджет *Карточка элемента данных* отображает сведения об одном элементе данных, позволяя быстро оценить его состояние и конфигурацию (последние данные, ошибки, описание и т. д.).

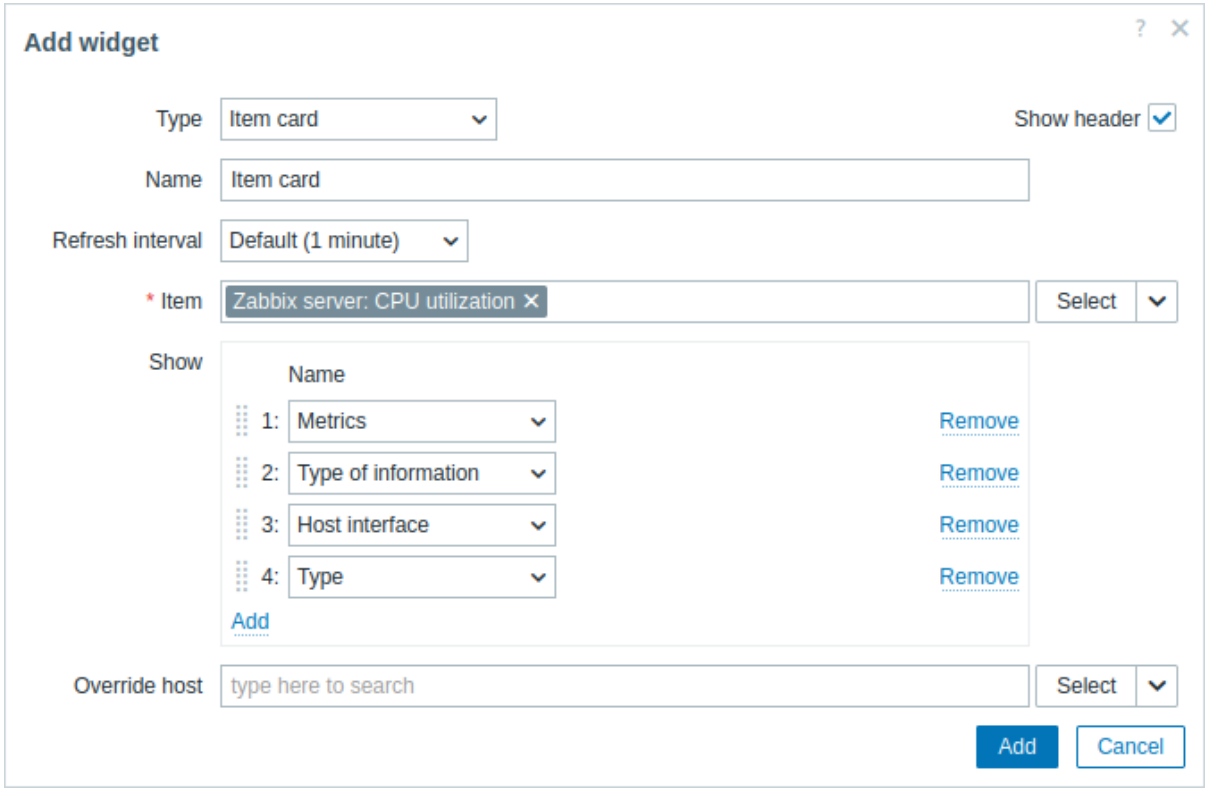


Элемент данных можно задать непосредственно в настройках виджета или выбрать из совместимого виджета на панели мониторинга.

Виджет может отображать несколько разделов, каждый из которых показывает различную информацию. Макет разделов автоматически подстраивается в зависимости от ширины виджета; при расширении по горизонтали разделы автоматически перераспределяются в несколько столбцов.

Настройка

Для настройки выберите тип *Карточка элемента данных*:



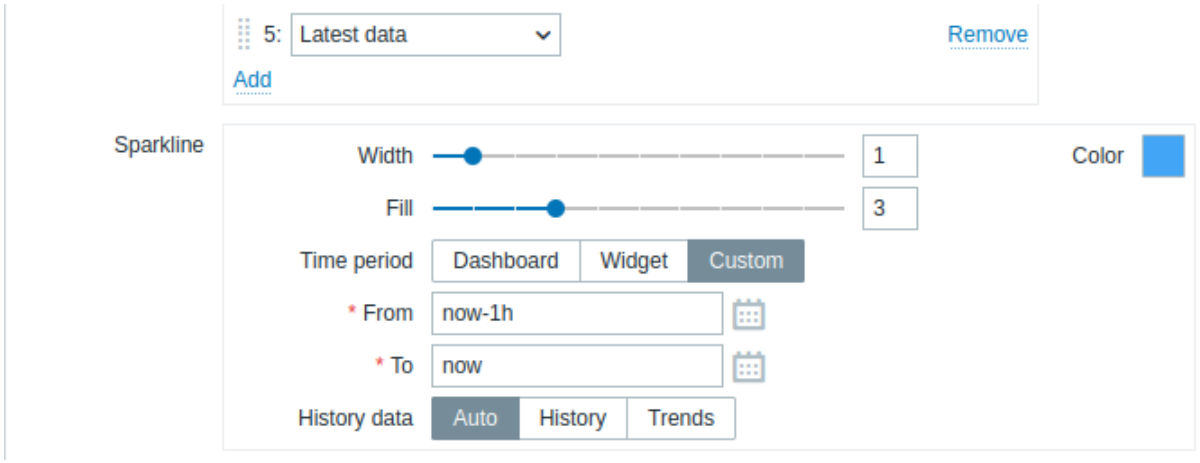
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Элемент данных	Выберите элемент данных. Либо выберите совместимый виджет в качестве источника данных для элементов данных. По мере ввода будут появляться подходящие варианты. Верхняя часть виджета всегда отображает **имя элемента данных** (при нажатии открывается меню элемента данных): - Если элемент данных отключен, рядом с именем элемента данных отображается красная метка *Отключено*. - В случае ошибок отображается квадратный значок с буквой "i". При наличии проблем для каждой важности проблемы отображается квадратный значок с количеством проблем. При наведении на значок открывается всплывающее окно с подробностями. - На панели шаблона отображается техническое имя элемента данных (вместо видимого имени). Верхняя часть виджета также отображает **путь элемента данных** — его узел сети и, если применимо, шаблон, правило low-level discovery (LLD) или основной элемент данных (нажатие на сущность открывает её настройки). Путь элемента данных отображается только пользователям, имеющим права доступа к соответствующим сущностям.
Показывать	Добавьте разделы для отображения в виджете. Разделы можно переупорядочить перетаскиванием.
Описание	Отображает описание элемента данных. Макросы раскрываются.
Текст ошибки	Отображает ошибку элемента данных (если есть).
Метрики	Отображает следующую информацию: - Интервал обновления элемента данных (кроме элементов данных trapper); - Настройка истории элемента данных (если есть); - Настройка трендов элемента данных (если есть). Макросы раскрываются. Если элемент данных использует пользовательские интервалы, рядом с интервалом обновления элемента данных отображается значок справки. При наведении на значок открывается всплывающее окно с подробностями о пользовательских интервалах.

Последние данные	Отображает следующую информацию: - Время с момента последней проверки элемента данных; - Последнее значение элемента данных; - Ссылку на простой график/историю значений элемента данных (если сохраняются история или тренды); - Настраиваемый график sparkline (если элемент данных числовой). Для элементов данных бинарного типа вместо последнего значения отображается миниатюра или опция *Показать*. При наведении на миниатюру открывается всплывающее окно с изображением. При наведении на *Показать* открывается всплывающее окно со значением элемента данных (строка Base64). На панели шаблона вместо последнего значения элемента данных отображается *Нет данных*.
Тип информации	Отображает тип информации элемента данных.
Триггеры	Отображает следующую информацию: - Метку *Триггеры* с количеством триггеров (при наведении на *Триггеры* открывается всплывающее окно с подробностями о триггерах); - Список имён триггеров (макросы раскрываются).
Интерфейс узла сети	Отображает интерфейс узла сети, используемый элементом данных, или *Нет данных*, если у элемента данных нет интерфейса узла сети (например, зависимые элементы данных, простые проверки). На панели шаблона отображается *Нет данных*.
Тип	Отображает тип элемента данных.
Инвентарные данные узла сети	Отображает поле инвентарных данных узла сети, которое заполняется значением элемента данных.
Теги	Отображает теги элемента данных.
Переопределить узел сети	Выберите совместимый виджет или селектор узла сети панели в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Спарклайн

Параметры настройки диаграммы спарклайна доступны только в том случае, если в виджет добавлен раздел *Последние данные*.



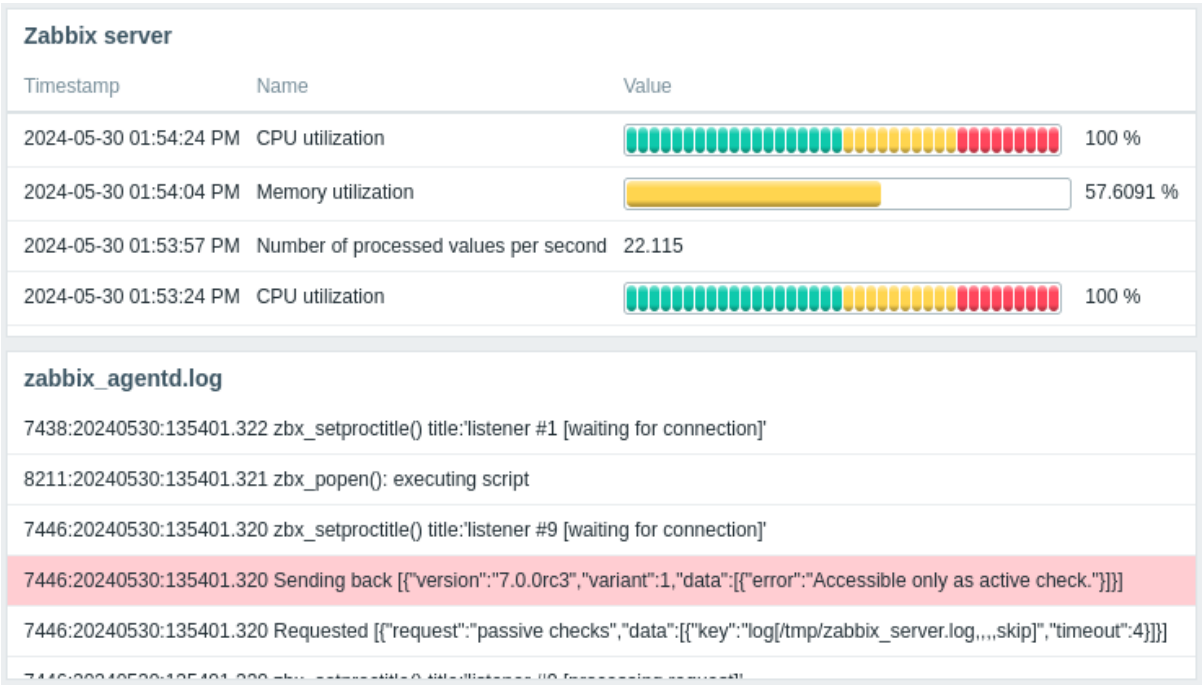
Ширина	Задайте толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Цвет	Выберите цвет линии и заливки.
Заливка	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.

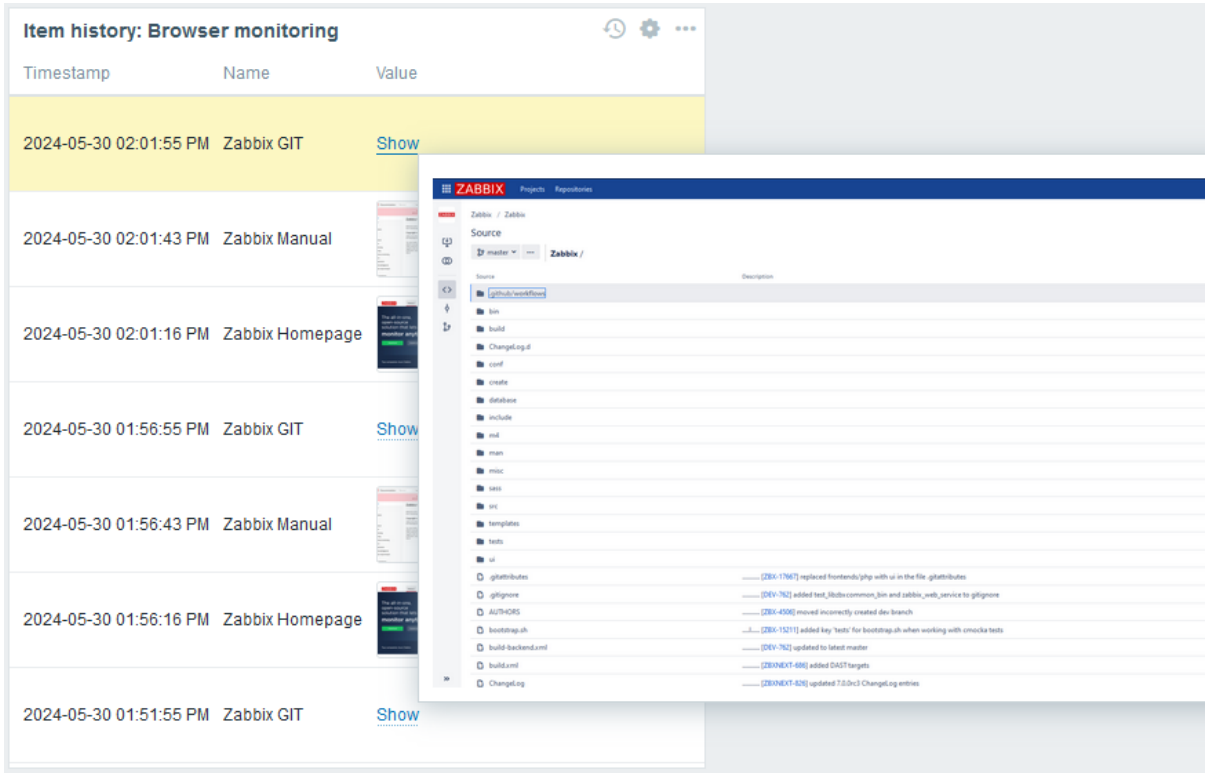
Период времени	Выберите источник данных для периода времени: **Панель** - использовать селектор периода времени панели; **Виджет** - использовать совместимый виджет (задается в параметре *Виджет*); **Пользовательский** - использовать пользовательский период времени, заданный в параметрах *От* и *До*; если задан, в правом верхнем углу виджета будет отображаться значок часов, указывающий установленное время при наведении курсора. Обратите внимание, что независимо от настройки *Период времени* виджета, совместимые виджеты все равно могут использовать его как источник данных для периода времени.
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Период времени* установлено значение "Виджет".
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Период времени* установлено значение "Пользовательский".
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Период времени* установлено значение "Пользовательский".
Данные истории	Брать данные из истории или трендов: **Авто** - автоматический выбор; **История** - брать данные истории; **Тренды** - брать данные трендов. Этот параметр применяется только к числовым данным. Нечисловые данные всегда будут браться из истории.

История элемента данных

Обзор

Виджет *Item history* отображает последние данные для всех типов элементов данных (числовых, текстовых и т. д.) в табличном формате. Он может показывать индикаторы прогресса и изображения (полезно для browser items) и может выделять значения (полезно для мониторинга файлов журнала).

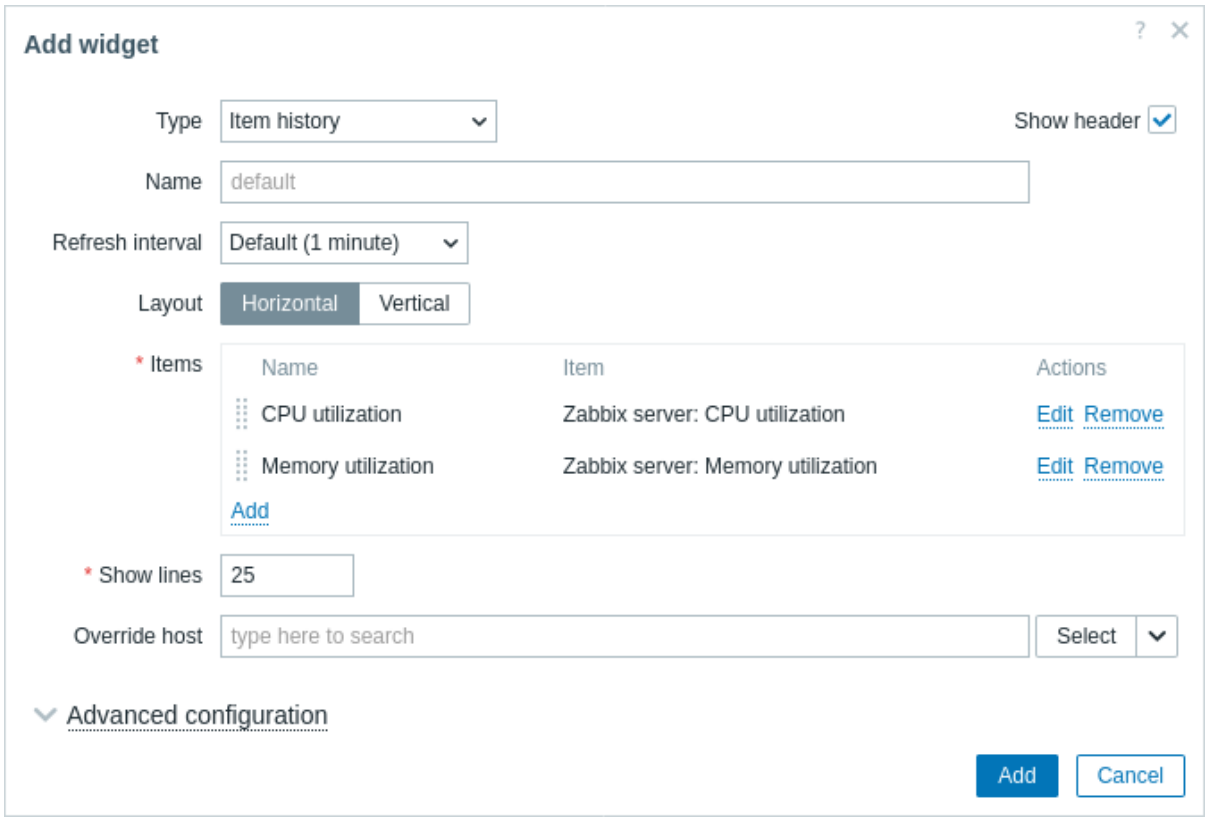




Можно отображать до 1000 записей.

Конфигурация

Чтобы настроить, выберите *Item history* в качестве типа:



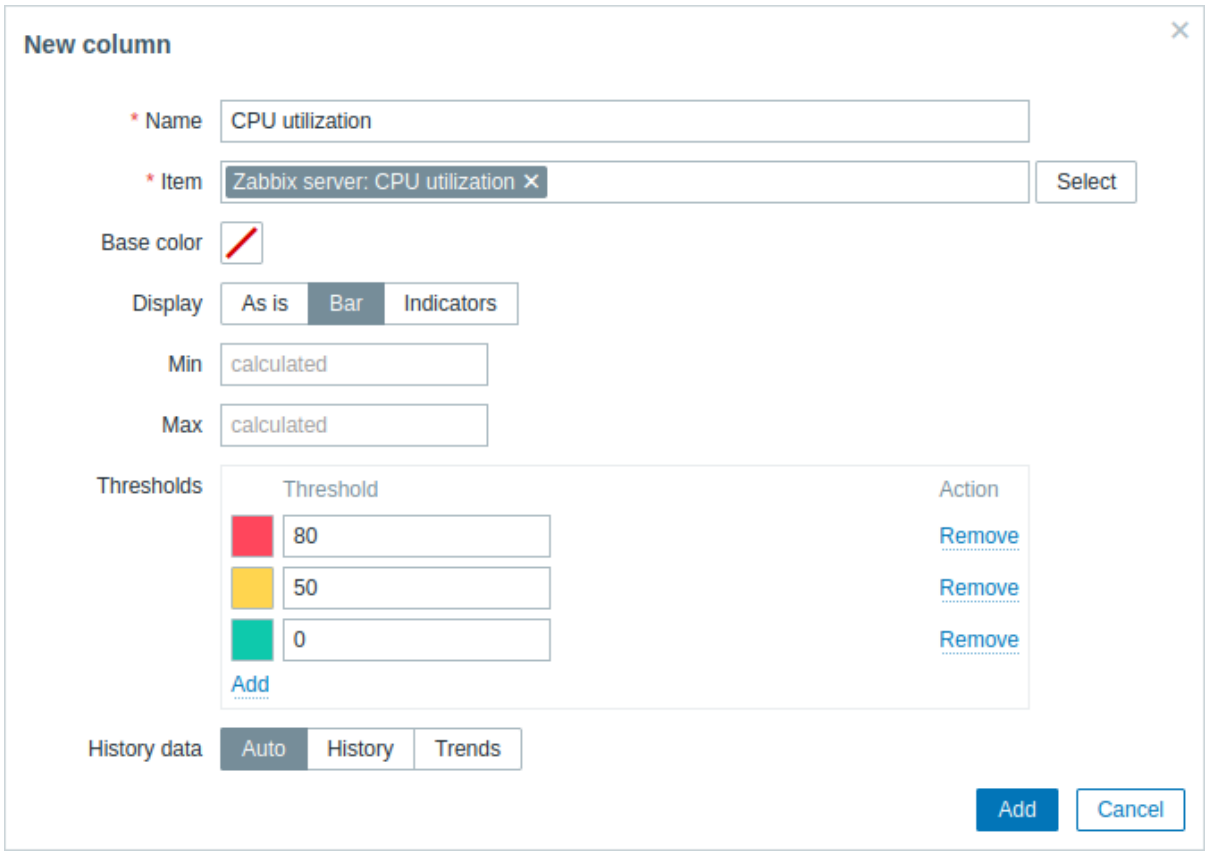
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Layout	Выберите вариант расположения столбцов элементов данных: **Horizontal** - элементы данных будут отображаться горизонтально, значения - вертикально; **Vertical** - элементы данных будут отображаться вертикально, значения - горизонтально.

Items	Добавьте столбцы элементов данных для отображения. Порядок элементов определяет порядок их отображения. Элементы можно менять местами, перетаскивая их вверх или вниз за маркер перед именем элемента.
Show lines	Укажите количество строк значений элемента данных для отображения.
Override host	Выберите совместимый виджет или селектор узла сети на панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Advanced configuration	Нажмите на метку *Advanced configuration*, чтобы отобразить параметры расширенной настройки.

Конфигурация столбцов

Чтобы настроить столбцы элементов данных, нажмите *Add* в параметре *Items*:



Общие параметры колонок:

Имя (Name)	Введите имя колонки. Если оставить пустым, используется имя элемента данных из параметра *Элемент данных (Item)*.
Элемент данных (Item)	Выберите элемент данных. Обратите внимание, что параметры настройки колонки отличаются в зависимости от типа информации выбранного элемента данных; для более подробных сведений смотрите отдельные параметры ниже. При настройке виджета на панели шаблона могут быть выбраны только элементы данных, настроенные в этом шаблоне.
Основной цвет (Base color)	Выберите цвет фона колонки или цвет заливки, если параметр *Отображение (Display)* имеет значения «Столбцы (Bar)» или «Индикатор (Indicators)». Обратите внимание, что основной цвет может быть переопределён цветом порога или подсветки.

Параметры колонок, специфичные для числовых типов элементов данных:

Отображение (Display)	Выберите, как следует отображать значение элемента данных: **Как есть (As is)** — в виде обычного текста; **Столбцы (Bar)** — в виде сплошной цветной полосы; **Индикаторы (Indicators)** — в виде сегментированной цветной полосы.
Мин (Min)	Введите минимальное значение для полосы/индикаторов. Если оставить пустым, виджет будет использовать минимальное значение элемента данных. Этот параметр доступен, только когда параметр *Отображение (Display)* имеет значения «Столбцы (Bar)» или «Индикаторы (Indicators)».
Макс (Max)	Введите максимальное значение для полосы/индикаторов. Если оставить пустым, виджет будет использовать максимальное значение элемента данных. Этот параметр доступен, только когда параметр *Отображение (Display)* имеет значения «Столбцы (Bar)» или «Индикаторы (Indicators)».
Пороги (Thresholds)	Нажмите *Добавить (Add)* для добавления порога, выберите цвет порога из палитры и укажите числовое значение. При сохранении список порого будет отсортирован по возрастанию. Поддерживаются суффиксы (например, «1d», «2w», «4K», «8G»). Поддерживаются преобразования значений.
Данные истории (History data)	Выберите, брать данные из истории или из динамики изменений: **Автоматически (Auto)** — автоматический выбор; **История (History)** — брать данные истории; **Динамика изменений (Trends)** — брать данные динамики изменений.

Параметры столбцов, специфичные для элементов данных типов character, text, log и JSON:

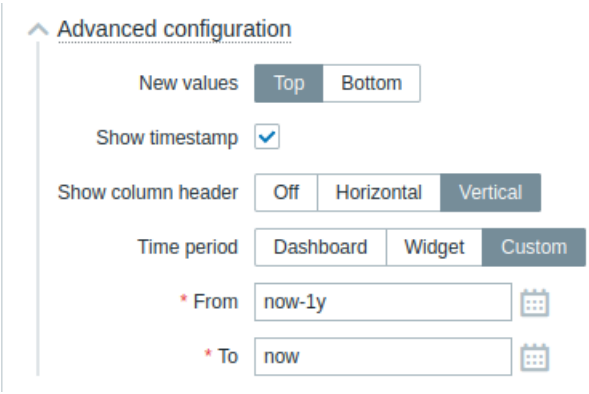
Подсветки	Нажмите *Добавить*, чтобы добавить подсветку, выберите цвет подсветки в палитре цветов и укажите регулярное выражение. Выбранный цвет будет использоваться как цвет фона для значений элемента данных, где указанное регулярное выражение совпадает с текстом.
Отображение	Выберите, как должно отображаться значение элемента данных: **Как есть** - точно в том виде, в котором оно получено. Разрывы строк сохраняются, если значение содержит символы перевода строки. Перенос слов применяется, если для *Макет* задано значение ”Вертикально” и настроено более одного столбца; **HTML** - как текст, отформатированный в HTML; **Одна строка** - как одна строка, сокращенная до указанной длины (1-500 символов). При наведении курсора или щелчке по сокращенному значению открывается всплывающее окно с полным значением.
Использовать моноширинный шрифт	Отметьте этот флажок, чтобы отображать значение элемента данных моноширинным шрифтом (по умолчанию не отмечен).
Показывать локальное время	Отметьте этот флажок, чтобы отображать локальное время вместо временной метки в столбце временной метки. Обратите внимание, что флажок *Показывать временную метку* в расширенной настройке также должен быть отмечен. Этот параметр доступен только для элементов данных типа log.

Параметры колонок, специфичные для элементов данных бинарного типа:

Миниатюра (Show thumbnail)	Установите этот флажок для отображения миниатюры для бинарных значений с изображениями или параметром «Показать (Show)» для бинарных значений, не являющихся изображениями. Снимите этот флажок для отображения параметра «Показать (Show)» для всех бинарных значений элементов данных. При наведении курсора мыши или нажатии на кнопку «Показать (Show)» открывается всплывающее окно со значением элемента данных (изображением или строкой Base64). Если значение элемента данных является пустой строкой, параметр «Показать (Show)» отображается; при наведении курсора мыши или нажатии открывается всплывающее окно, содержащее «Пустая строка» (”Empty string”).

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка*:

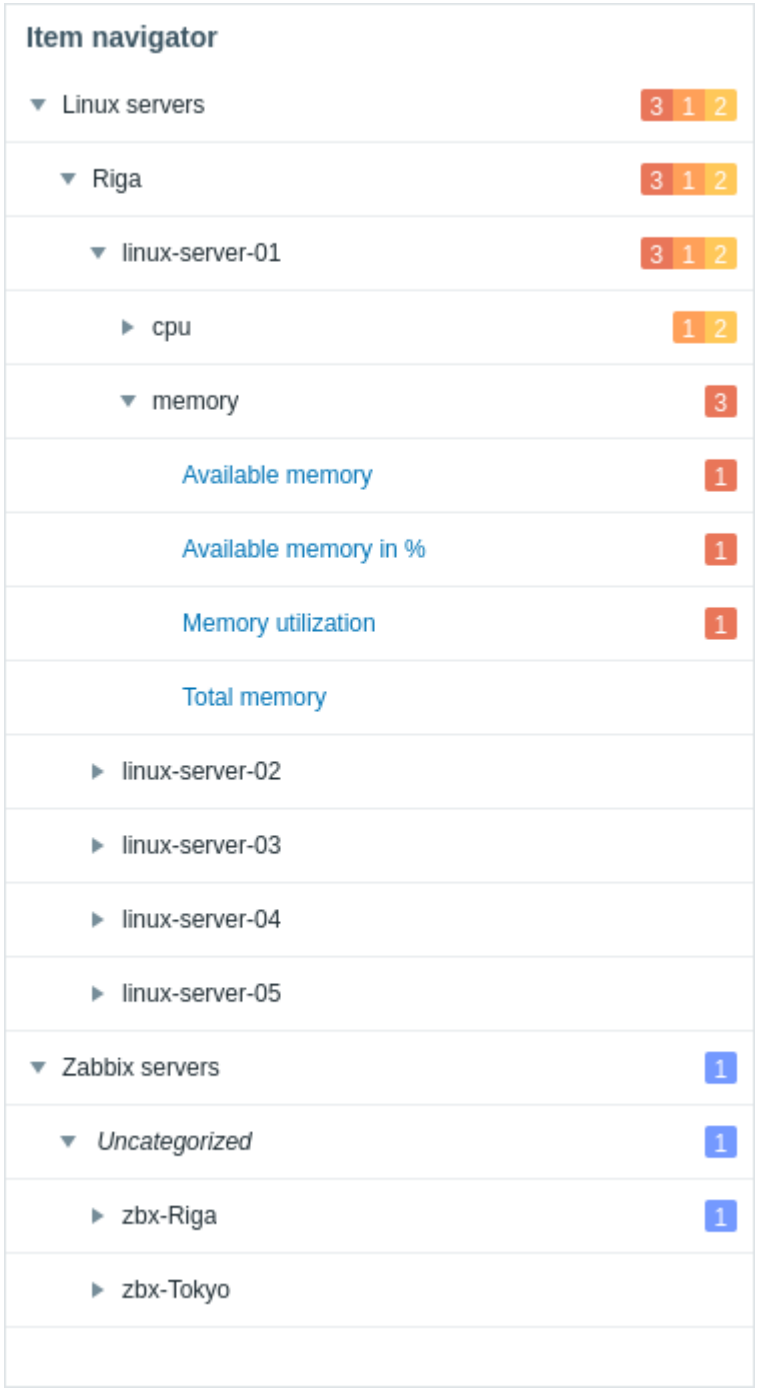


Новые значения	Выберите, куда следует добавлять новые значения элемента данных: **Top** - в верхнюю часть столбцов; **Bottom** - в нижнюю часть столбцов.
Показывать временную метку	Установите этот флажок, чтобы отображать столбец временной метки (по умолчанию не установлен).
Показывать заголовок столбца	Выберите ориентацию заголовка столбца: **Off** - скрыть заголовок; **Horizontal** - отображать заголовок горизонтально; **Vertical** - отображать заголовок вертикально.
Период времени	Выберите источник данных для периода времени: **Dashboard** - использовать селектор периода времени панели; **Widget** - использовать совместимый виджет (указанный в параметре *Widget*); **Custom** - использовать пользовательский период времени, заданный в параметрах *From* и *To*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, указывающий заданное время при наведении курсора мыши. Обратите внимание, что независимо от настройки *Период времени* виджета, совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.
Widget	Введите или выберите совместимый виджет (*Graph*, *Graph (classic)*, *Graph prototype*) в качестве источника данных для периода времени. Этот параметр доступен, если для *Период времени* установлено значение "Widget".
From	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Период времени* установлено значение "Custom".
To	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Период времени* установлено значение "Custom".

Навигатор элементов данных

Обзор

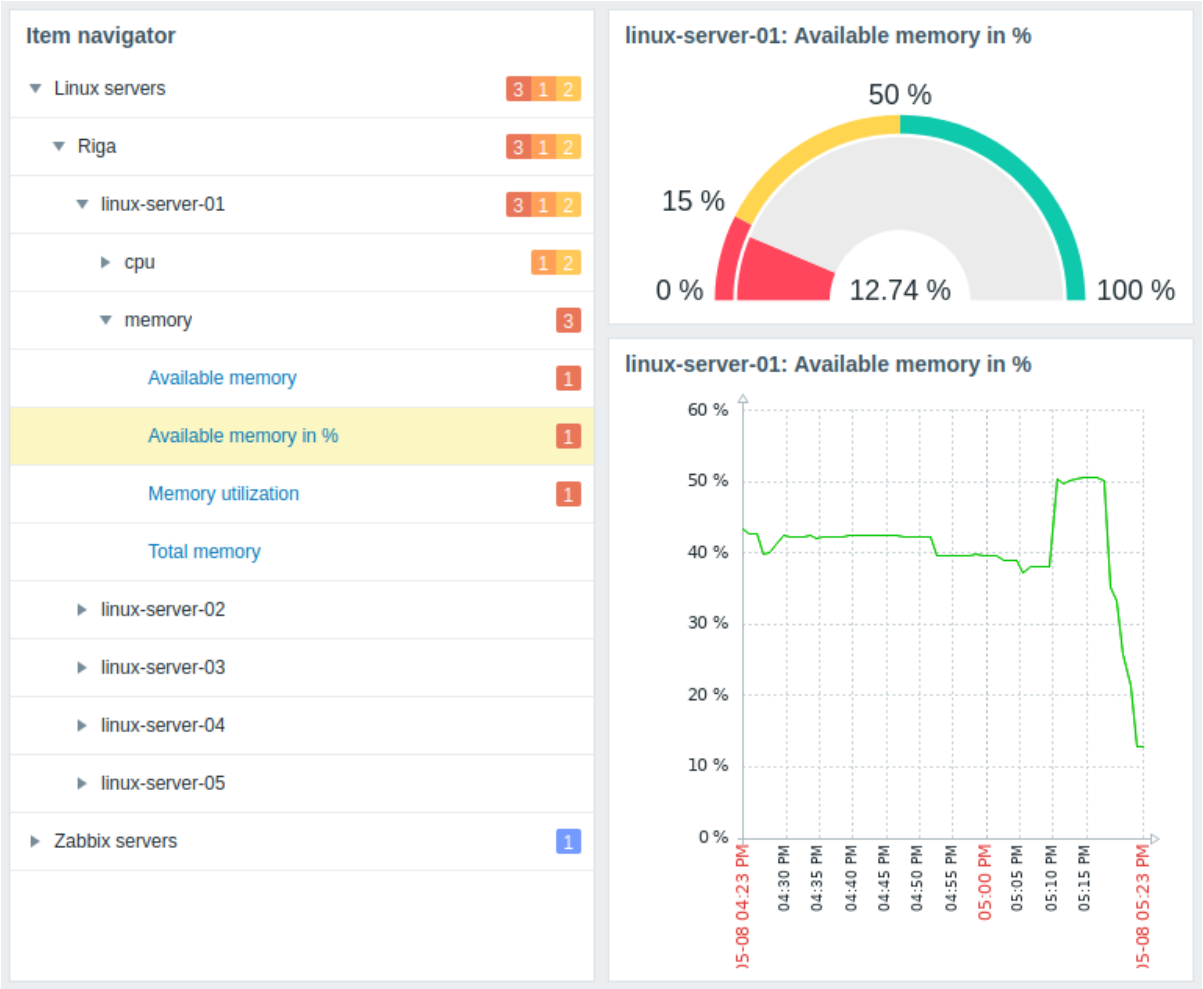
Виджет *Item navigator* отображает список элементов данных на основе различных параметров фильтрации и группировки.



Группы, по которым организованы элементы данных, можно разворачивать или сворачивать.

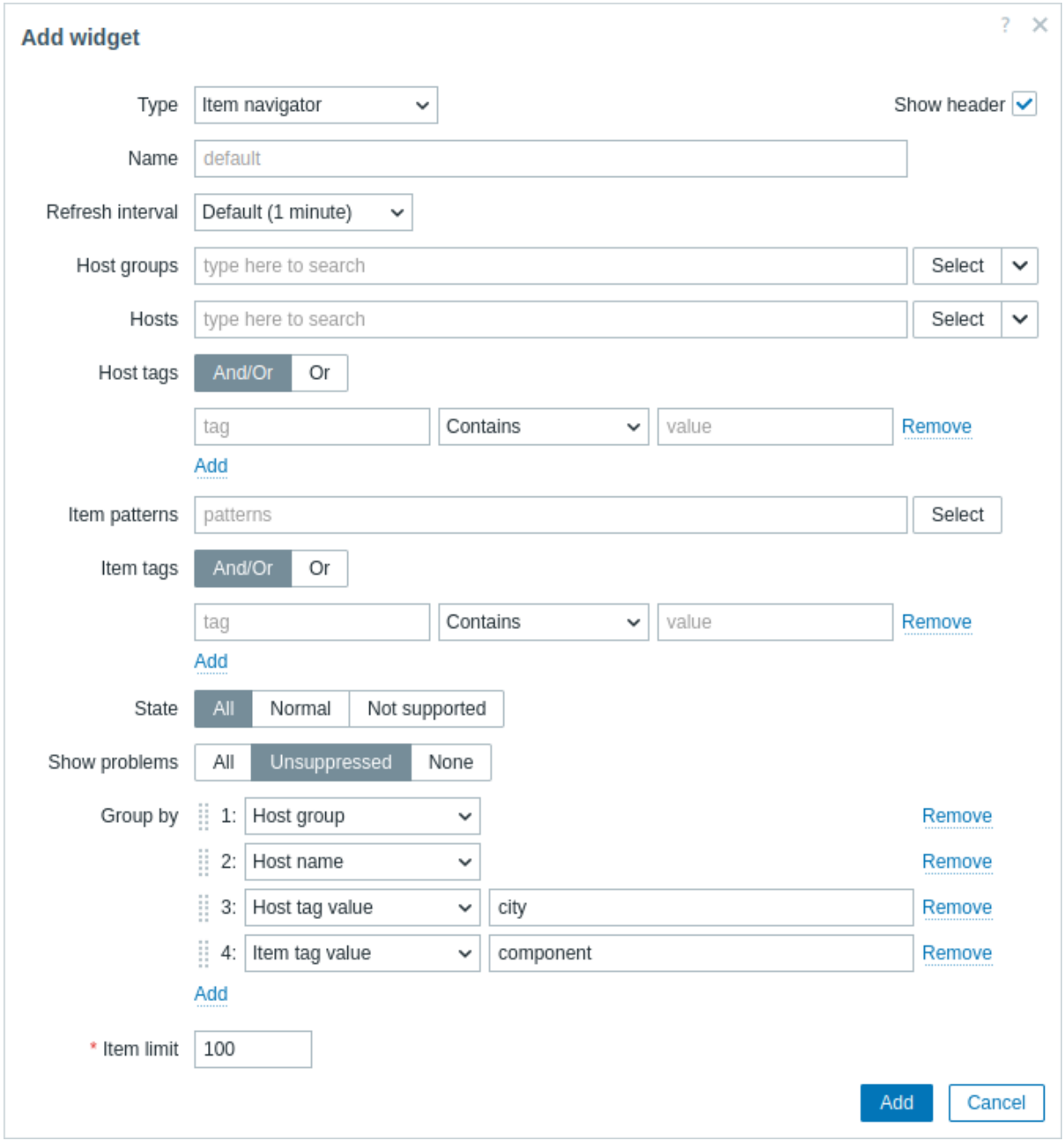
Для групп и проблем дополнительные сведения доступны во всплывающих подсказках при наведении мыши.

Этот виджет особенно полезен для управления тем, что отображают другие виджеты, на основе выбранного элемента данных.



Конфигурация

Для настройки выберите тип *Навигатор элементов данных*:



Помимо параметров, общих для всех виджетов, можно задать следующие специфические параметры:

Группы узлов сети	Выберите группы узлов сети. Также можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение: после начала ввода имени группы появится раскрывающийся список соответствующих групп. При выборе родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети; если группы узлов сети не выбраны, виджет отображает элементы данных, принадлежащие всем узлам сети из всех групп узлов сети. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.
Узлы сети	Выберите узлы сети. Также можно выбрать совместимый виджет или панель мониторинга в качестве источника данных для узлов сети. Это поле поддерживает автодополнение: после начала ввода имени узла сети появится раскрывающийся список соответствующих узлов сети. Если узлы сети не выбраны, виджет отображает элементы данных, принадлежащие всем узлам сети. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.

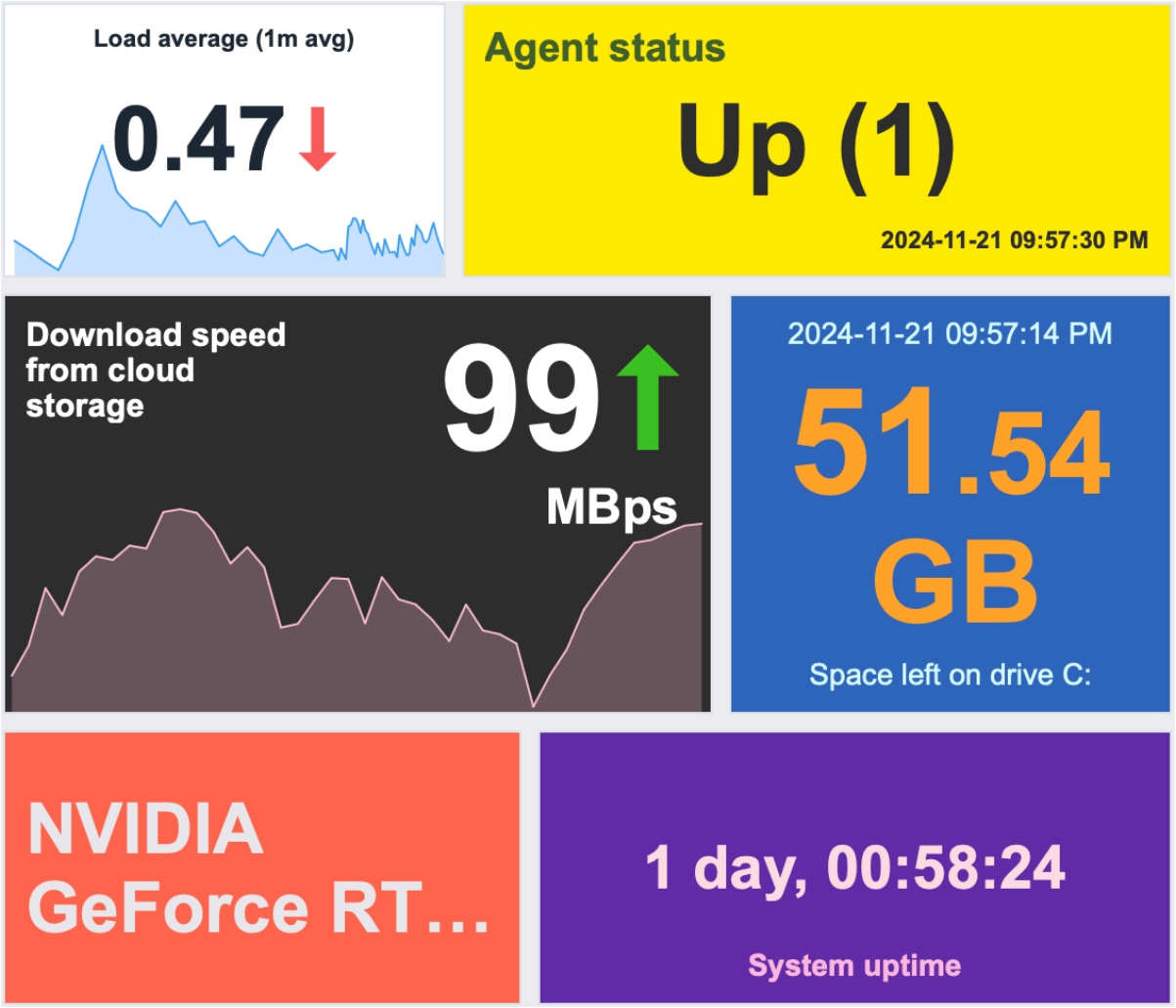
Теги узлов сети	Укажите теги узлов сети для фильтрации элементов данных, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега объединяются оператором ИЛИ; **ИЛИ** - достаточно выполнения одного условия.
Шаблоны элементов данных	Этот параметр недоступен при настройке виджета на панели мониторинга шаблона. Введите шаблоны элементов данных или выберите существующие элементы данных в качестве шаблонов. На навигаторе элементов данных будут отображаться элементы данных, соответствующие указанным шаблонам. Для выбора можно использовать шаблоны с подстановочными символами (например, `*` возвращает элементы данных, соответствующие нулю или более символам; `Zabbix*` возвращает элементы данных, начинающиеся с "Zabbix"). Чтобы указать шаблон с подстановочным символом, введите строку вручную и нажмите *Enter*. При начале ввода в раскрывающемся списке отображаются соответствующие элементы данных, ограниченные элементами данных выбранных *Узлов сети* или узлов сети из выбранных *Групп узлов сети*, если они указаны. Символ подстановки интерпретируется всегда, поэтому невозможно добавить, например, элемент данных с именем *item** отдельно, если существуют другие соответствующие элементы данных (например, item2, item3). При настройке виджета на панели мониторинга шаблона этот параметр позволяет выбирать только элементы данных, настроенные в шаблоне.
Теги элементов данных	Укажите теги элементов данных для фильтрации элементов данных, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега объединяются оператором ИЛИ; **ИЛИ** - достаточно выполнения одного условия.
Состояние	Фильтруйте отображаемые элементы данных по их состоянию (все, нормальные, не поддерживаются).

Показывать проблемы	Фильтруйте отображаемые проблемы по их статусу (все, неподтвержденные, отсутствуют). Проблемы обозначаются цветными блоками в конце строки элемента данных. Цвета зависят от серьезности проблемы, которую можно настроить в разделе *Администрирование* > *Общие* > *Параметры отображения триггеров*. Серьезность проблем можно изменить при обновлении проблем.
Группировать по	Добавьте атрибут группировки, по которому будут группироваться элементы данных: **Группа узлов сети** - группировать элементы данных по группам узлов сети, к которым относятся их узлы сети; **Имя узла сети** - группировать элементы данных по узлам сети; **Значение тега узла сети** - введите имя тега, чтобы группировать элементы данных по значениям этого тега узла сети (например, введите ”city”, чтобы группировать элементы данных по значениям ”Riga”, ”Tokyo” и т. д.); **Значение тега элемента данных** - введите имя тега, чтобы группировать элементы данных по значениям этого тега элемента данных (например, введите ”component”, чтобы группировать элементы данных по значениям ”cpu”, ”memory” и т. д.). Атрибуты группировки можно менять местами, перетаскивая их вверх или вниз за маркер перед именем группы. Обратите внимание, что порядок атрибутов группировки определяет порядок вложенности групп. Например, указание нескольких имен тегов узлов сети (1: color, 2: city) приведет к группировке элементов данных сначала по цвету (red, blue и т. д.), а затем по городу (Riga, Tokyo и т. д.). Элемент данных может отображаться в нескольких группах в зависимости от настроенных атрибутов группировки (например, при группировке по группе узлов сети, если узел сети элемента данных принадлежит нескольким группам узлов сети). При нажатии на такой элемент данных он выбирается и выделяется во всех группах. Элементы данных, не соответствующие настроенным атрибутам группировки, отображаются в группе *Без категории*. Если параметр *Показывать проблемы* настроен на отображение проблем, они отображаются следующим образом: - для каждой группы отображается общее количество проблем во вложенных элементах данных; - для каждого элемента данных отображается только количество его проблем. Можно указать до 10 атрибутов группировки, и все они должны быть уникальными. Если атрибуты группировки не указаны, элементы данных не группируются.
Ограничение количества элементов данных	Введите максимальное количество отображаемых элементов данных. Допустимые значения находятся в диапазоне от 1 до 9999. Если доступно больше элементов данных, чем указано в ограничении, под отображаемыми элементами данных появляется соответствующее сообщение (например, ”Отображается 100 из 100+ элементов данных”). Обратите внимание, что настроенное ограничение количества элементов данных также влияет на отображение настроенных групп; например, если ограничение установлено равным 100, а элементы данных сгруппированы по узлам сети (каждый из которых содержит 200 элементов данных), в виджете будет отображаться только первый узел сети с его 100 элементами данных. На этот параметр не влияет параметр *Ограничение для результатов поиска и фильтрации* в разделе *Администрирование* > *Общие* > *Веб-интерфейс*.

Значение элемента данных

Обзор

Виджет *Item value* отображает значение одного числового или строкового элемента данных. Он полезен для отслеживания важных метрик, визуализации пороговых значений и обнаружения резких изменений в данных.



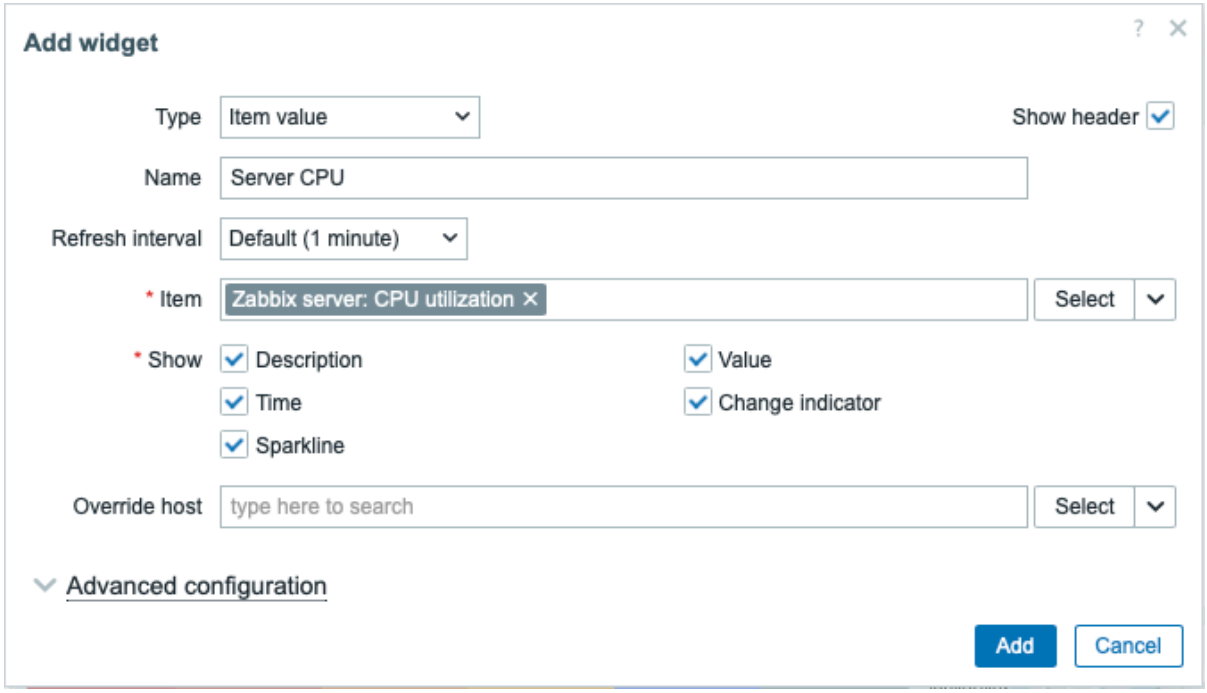
Вы можете настроить виджет так, чтобы он отображал следующее:

- Временную метку метрики (2024-11-21 09:57:30)
- Описание элемента данных (Load average (1m), Agent status)
- Значение элемента данных, индикатор изменения и единицы измерения (0.47 ↓, 99 ↑ Mbps)
- Цвет фона
- График sparkline для значений за указанный период времени

При нажатии на виджет открывается ad-hoc graph для числовых элементов данных или последние данные для строковых элементов данных.

Конфигурация

Чтобы настроить, выберите *Item value* в качестве типа виджета:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Item	Выберите элемент данных. Либо выберите совместимый виджет в качестве источника данных для элемента данных. Элементы данных, возвращающие двоичные данные, не поддерживаются.
Show	Установите флажок, чтобы отображать соответствующий элемент: описание, значение, время, индикатор изменения, sparkline. Снимите флажок, чтобы скрыть. Должен быть выбран как минимум один элемент.
Override host	Выберите совместимый виджет или селектор узла сети на панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Advanced configuration	Нажмите заголовок *Advanced configuration*, чтобы отобразить параметры дополнительной настройки.

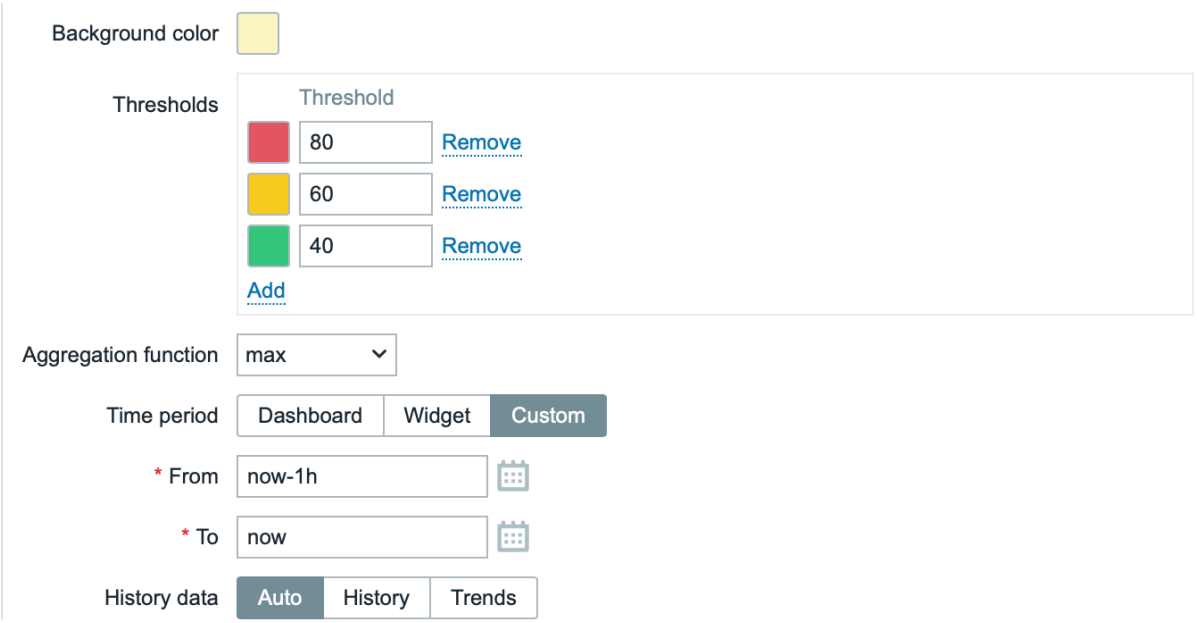
Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка* и только для тех элементов, которые выбраны в поле *Show* (см. выше).

Обратите внимание, что несколько элементов не могут занимать одно и то же пространство; если они размещены в одном пространстве, будет отображено сообщение об ошибке.

Общие параметры виджета

Эти параметры определяют цвет фона (статический или динамический) для всего виджета и функцию агрегации для отображения значений.

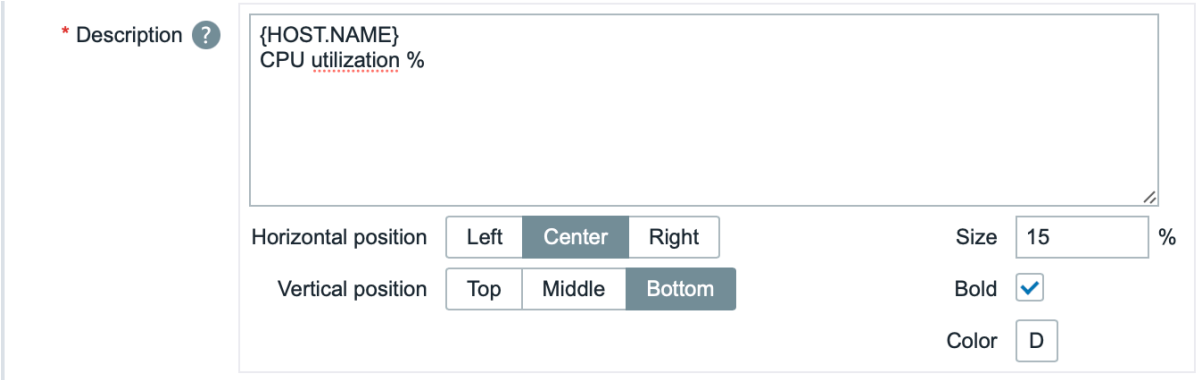


Цвет фона	Выберите цвет фона для всего виджета в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать значение по умолчанию* в палитре цветов.
Пороговые значения	Настройте динамический цвет фона для всего виджета. Нажмите *Добавить*, чтобы добавить пороговое значение, выберите цвет фона в палитре цветов и укажите числовое значение. Когда значение элемента данных становится равным пороговому значению или превышает его, цвет фона изменяется. При сохранении список будет отсортирован по возрастанию. Обратите внимание, что динамический цвет фона корректно отображается только для числовых элементов данных.
Функция агрегации	Укажите функцию агрегации, которую следует использовать в пределах выбранного *Периода времени*: **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **count** - отображать количество значений; **sum** - отображать сумму значений; **first** - отображать первое значение; **last** - отображать последнее значение. Если выбрано **не используется**, отображается самое последнее значение (без агрегации). Агрегация позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день) вместо самого последнего значения. Для функций *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для функции *count* нечисловые данные будут преобразованы в числовые.
Период времени	Период времени, используемый для агрегации значений. Выберите источник данных для периода времени: **Панель мониторинга** - использовать селектор периода времени панели мониторинга; **Виджет** - использовать совместимый виджет (указанный в параметре *Виджет*); **Пользовательский** - использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении указателя мыши. Обратите внимание, что независимо от настройки *Период времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени. Этот параметр скрыт, если для *Функции агрегации* установлено значение «не используется».
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Периода времени* установлено значение «Виджет».

От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Периода времени* установлено значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Периода времени* установлено значение «Пользовательский».
Данные истории	Получать данные из истории или трендов: **Авто** - автоматический выбор; **История** - получать данные истории; **Тренды** - получать данные трендов. Эта настройка применяется только к числовым данным. Нечисловые данные всегда будут получаться из истории.

Описание

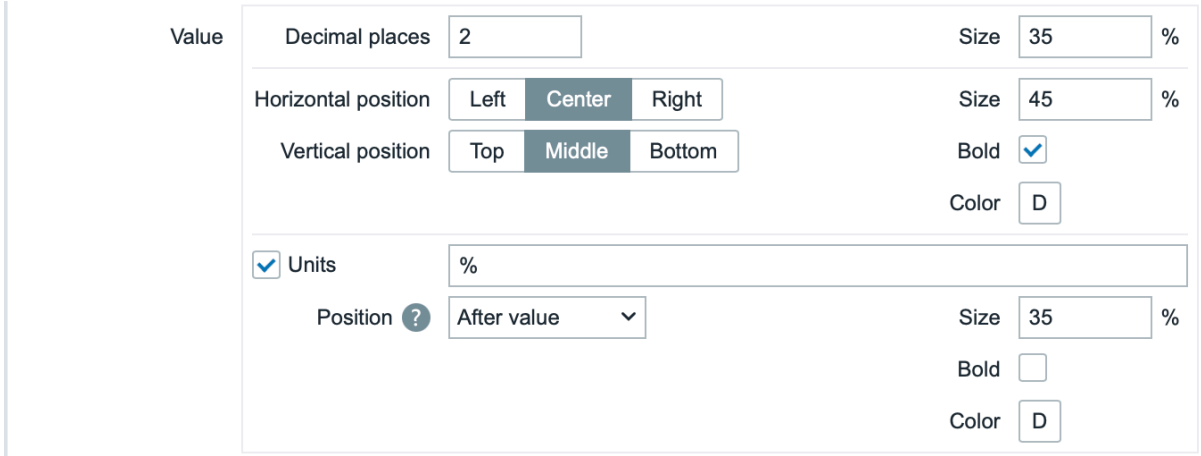
Эти параметры определяют, как должно отображаться описание элемента данных.



Описание	Введите описание элемента данных. Это описание может переопределить имя элемента данных по умолчанию. Поддерживаются многострочные описания. Возможно сочетание текста и поддерживаемых макросов. {HOST.*}, {ITEM.*}, {INVENTORY.*} и пользовательские макросы поддерживаются.
Горизонтальное положение	Выберите горизонтальное положение описания элемента данных — слева, справа или по центру.
Вертикальное положение	Выберите вертикальное положение описания элемента данных — сверху, снизу или посередине.
Размер	Введите высоту размера шрифта для описания элемента данных (в процентах относительно общей высоты виджета).
Жирный	Установите этот флажок, чтобы отображать описание элемента данных жирным шрифтом.
Цвет	Выберите цвет описания элемента данных в палитре цветов. `D` обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Use default* в палитре цветов.

Значение

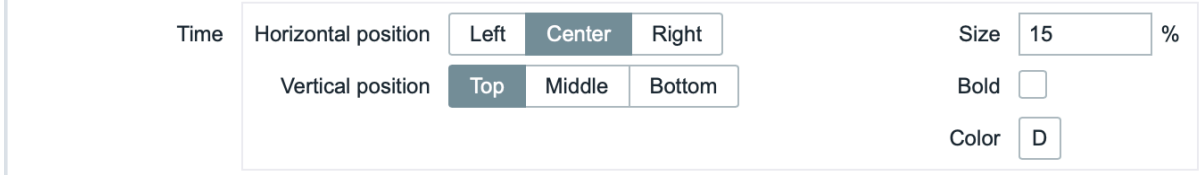
Эти параметры определяют, как должно отображаться значение элемента данных.



Десятичные знаки	Выберите, сколько знаков после запятой будет отображаться со значением. Это значение влияет только на элементы данных с плавающей точкой. Для элементов данных, у которых в качестве единиц измерения указано ”s”, если это значение больше 0 (по умолчанию 2), виджет округляет наиболее значимую единицу времени и отображает числовое значение (например, ”10.43m”) вместо преобразования значения в полную строку времени. Если установлено 0, секунды преобразуются в удобочитаемую строку времени (например, ”4h 56m 30s”).
Размер	Введите высоту размера шрифта для десятичных знаков (в процентах относительно общей высоты виджета).
Горизонтальное положение	Выберите горизонтальное положение значения элемента данных — слева, справа или по центру.
Вертикальное положение	Выберите вертикальное положение значения элемента данных — сверху, снизу или посередине.
Размер	Введите высоту размера шрифта для значения элемента данных (в процентах относительно общей высоты виджета). Обратите внимание, что размер значения элемента данных имеет приоритет; остальные элементы должны уступать место значению. Однако для индикатора изменения, если значение слишком велико, оно будет обрезано, чтобы показать индикатор изменения.
Жирный	Установите флажок, чтобы отображать значение элемента данных жирным шрифтом.
Цвет	Выберите цвет значения элемента данных в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Единицы измерения	
Единицы измерения	Установите флажок, чтобы отображать единицы измерения вместе со значением элемента данных. Если вы введете название единицы измерения, оно заменит единицу измерения из конфигурации элемента данных.
Положение	Выберите положение единицы измерения элемента данных — над, под, перед или после значения.
Размер	Введите высоту размера шрифта для единицы измерения элемента данных (в процентах относительно общей высоты виджета).
Жирный	Установите флажок, чтобы отображать единицу измерения элемента данных жирным шрифтом.
Цвет	Выберите цвет единицы измерения элемента данных в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Use default* в палитре цветов.

Время

Эти параметры определяют, как должно отображаться время (значение часов из истории элемента данных).



Горизонтальное положение	Выберите горизонтальное положение времени — слева, справа или по центру.
Вертикальное положение	Выберите вертикальное положение времени — сверху, снизу или посередине.
Размер	Введите высоту размера шрифта для времени (в процентах относительно общей высоты виджета).
Жирный	Установите этот флажок, чтобы отображать время жирным шрифтом.
Цвет	Выберите цвет времени в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуться к значению по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.

Индикатор изменения

В этом разделе можно выбрать цвет индикаторов изменения с помощью палитры цветов.



Доступны следующие индикаторы изменения:

- ↑ - значение элемента данных увеличилось (для числовых элементов данных)
- ↓ - значение элемента данных уменьшилось (для числовых элементов данных)
- ↕ - значение элемента данных изменилось (для строковых элементов данных и элементов данных с преобразованием значений)

Примечание:

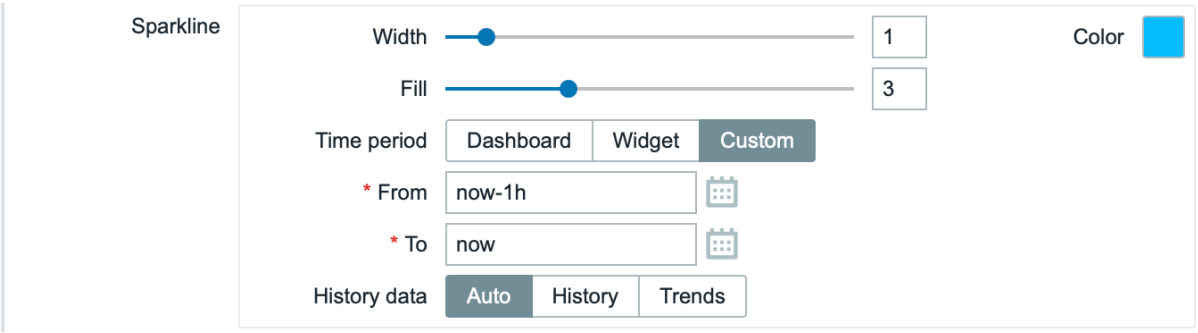
- Индикаторы увеличения (↑) и уменьшения (↓) не отображаются, если доступно только одно значение.
- Значения всегда сравниваются в пределах одной области; например:
  - Последнее значение сравнивается с предыдущим значением.
  - Значение за месяц сравнивается со значением за предыдущий месяц.
  - Для агрегированных значений предыдущий период имеет такую же продолжительность, как выбранный период, и заканчивается непосредственно перед началом выбранного периода.

D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы восстановить значение по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.

Вертикальный размер индикатора изменения равен размеру значения (целой части значения для числовых элементов данных).

Спарклайн

Эти параметры определяют, как должна отображаться диаграмма спарклайна.



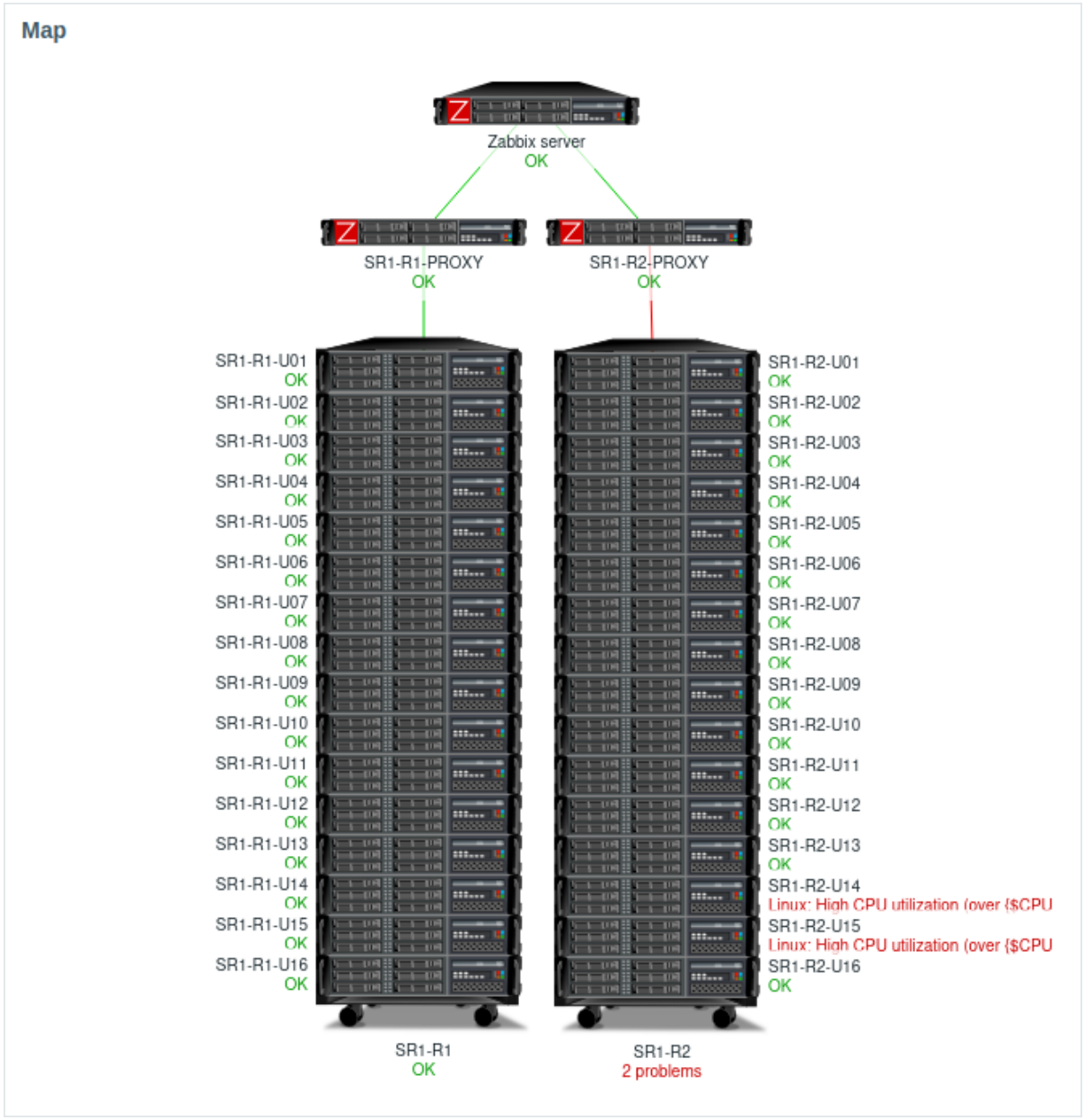
Ширина	Задайте толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Цвет	Выберите цвет линии и заливки.
Заливка	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.

Период времени	Укажите период времени, значения за который должны быть включены в диаграмму спарклайна. Выберите источник данных для периода времени: **Панель** - использовать селектор периода времени панели; **Виджет** - использовать совместимый виджет (задаётся в параметре *Виджет*); **Пользовательский** - использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, указывающий установленное время при наведении курсора. Обратите внимание, что независимо от настройки *Период времени* виджета, совместимые виджеты всё равно могут использовать его как источник данных для периода времени. Этот параметр недоступен, если *Функция агрегации* установлена в значение "не используется".
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если *Период времени* установлено в значение "Виджет".
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если *Период времени* установлено в значение "Пользовательский".
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если *Период времени* установлено в значение "Пользовательский".
Данные истории	Брать данные из истории или трендов: **Авто** - автоматический выбор; **История** - брать данные истории; **Тренды** - брать данные трендов.

Карта

Обзор

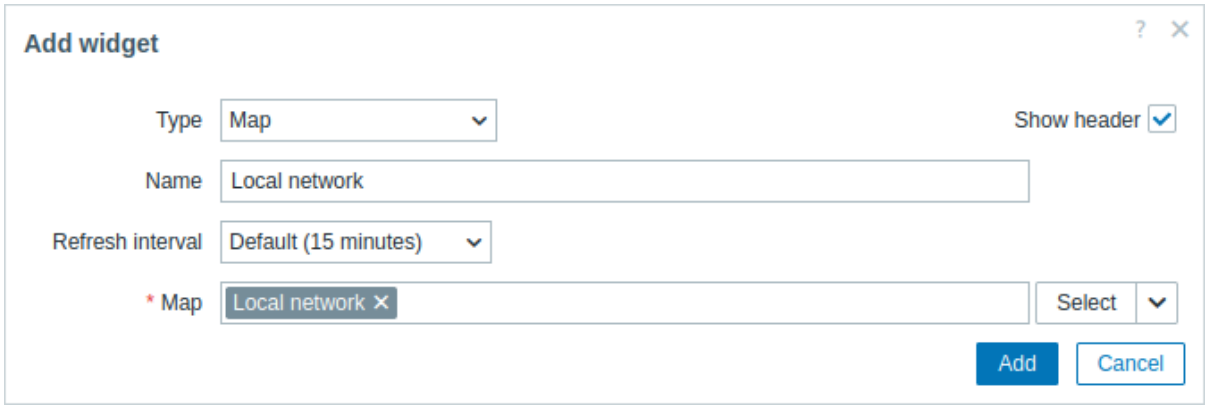
Виджет *Map* может отображать сетевую карту или аналогичную визуализацию, предоставляя вам динамический обзор вашей сети.



Вы можете использовать его вместе с виджетом Map navigation tree, чтобы динамически отображать карту, выбранную в дереве.

Настройка

Для настройки выберите *Карта сети (Map)* в качестве типа:



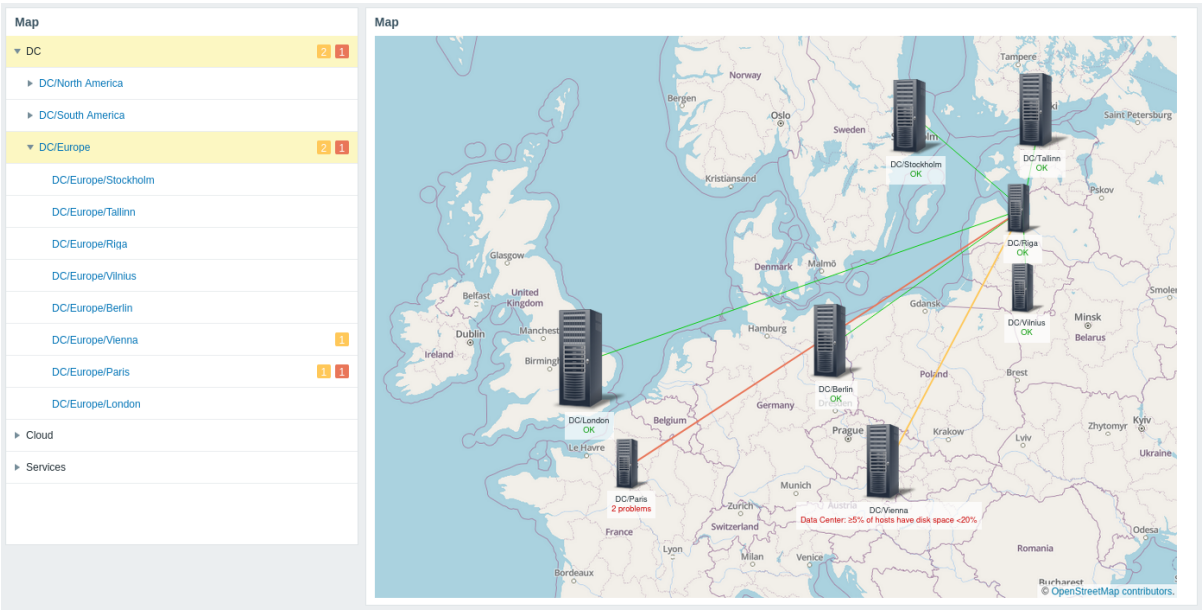
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете установить следующие специфичные опции:

Карта сети (Map)	Задайте карту для отображения. В качестве альтернативы выберите для отображаемой карты сети в качестве источника данных совместимый виджет. Это поле с функцией автодополнения; таким образом, после начала ввода имени карты или виджета будет автоматически предложен список совпадающих карт сети или виджетов в выпадающем меню.

Дерево навигации по карте

Обзор

Виджет *Map navigation tree* отображает иерархию существующих карт и количество проблем для каждой карты и группы карт.

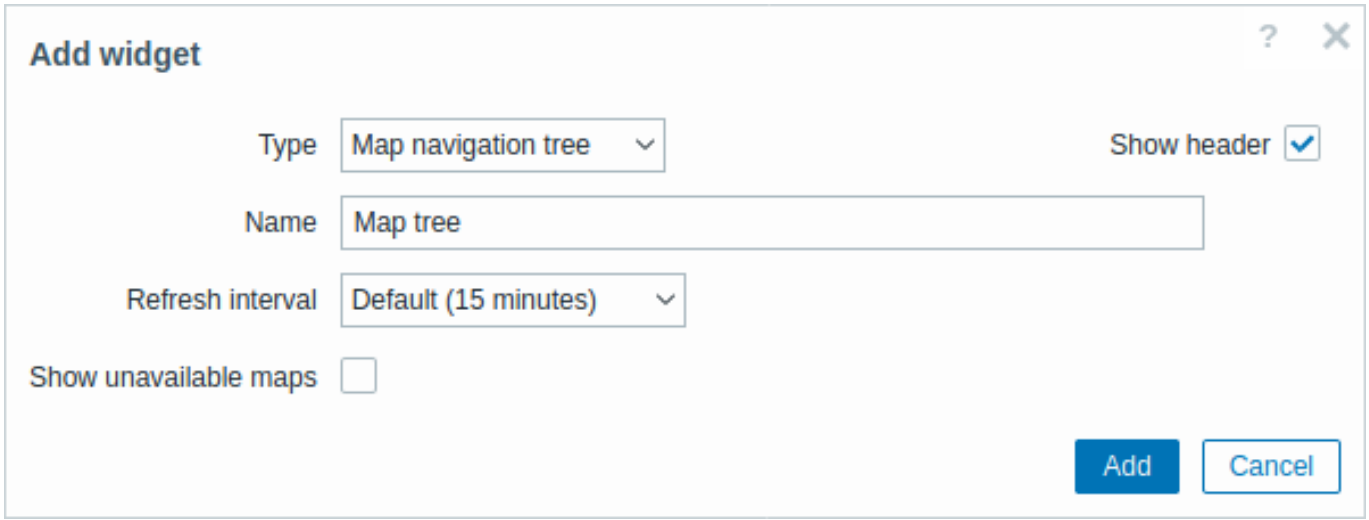


Его можно использовать вместе с виджетом Map, чтобы динамически отображать карту, выбранную в дереве.

Количество проблем для карты верхнего уровня в иерархии включает все проблемы подчиненных карт и собственные проблемы.

Настройка

Для настройки выберите *Дерево навигации карт сетей (Map navigation tree)* в качестве типа:

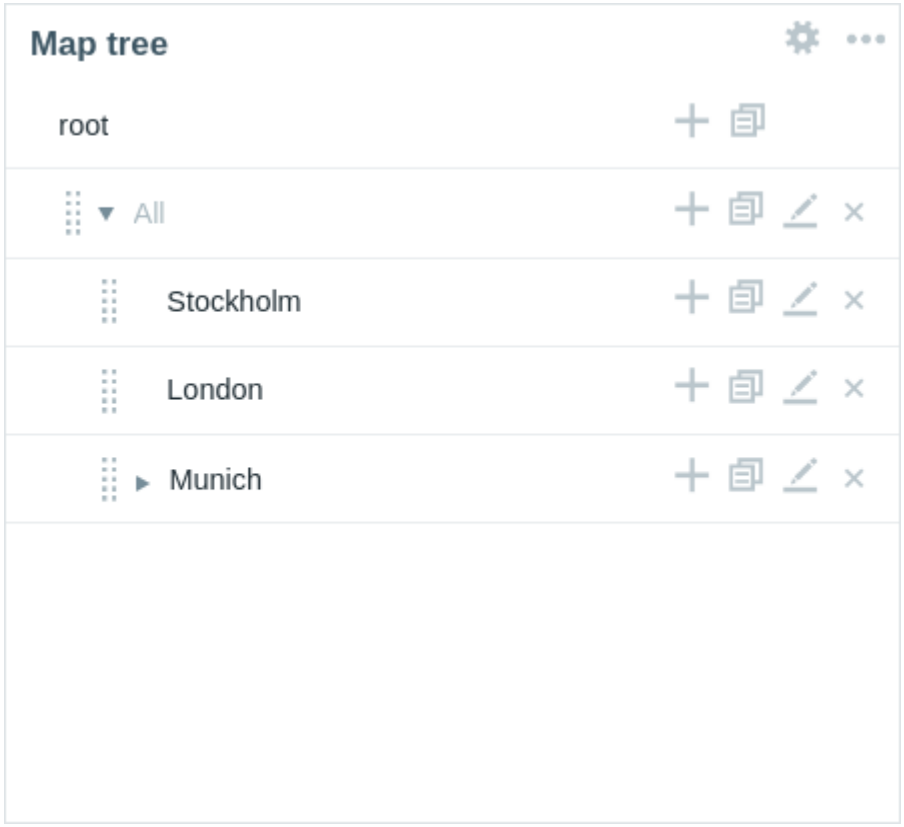


В дополнение к параметрам, которые являются общими для всех виджетов, вы можете выставить следующие специфичные опции:

Отображать недоступные карты (Show unavailable maps)	Установите этот флажок для отображения карт, на которые у пользователя нет прав на чтение. Недоступные карты в дереве навигации будут отображаться серым значком. Обратите внимание что если этот флажок отмечен, доступные карты нижнего уровня отображаются, даже если карта вышестоящего уровня недоступна. Если флажок снят, доступные карты нижних уровней при недоступности карт вышестоящих уровней вообще не будут отображаться. Количество проблем рассчитывается на основе доступных карт и доступных элементов карты.

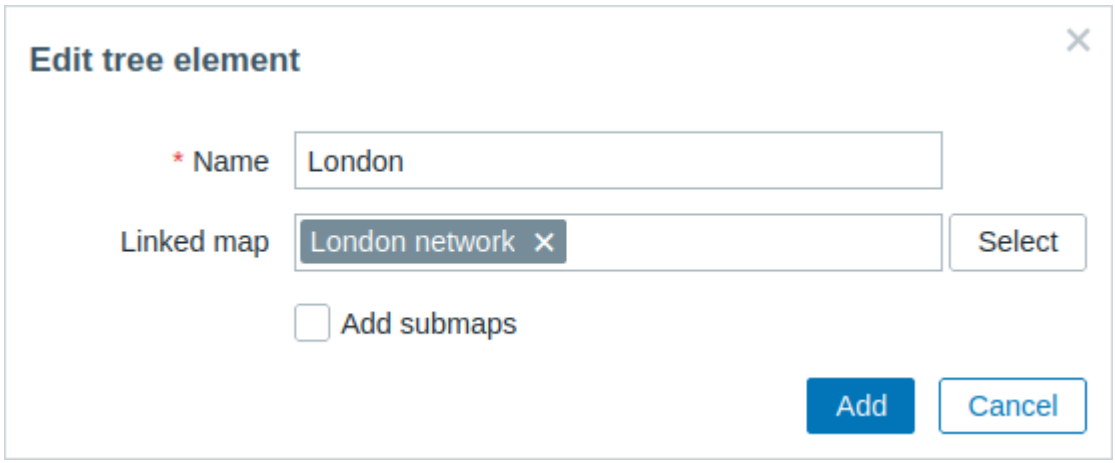
Элементы дерева навигации отображаются в виде списка. Вы можете:

- перетащить элемент (включая его дочерние элементы) на новое место в списке;
- развернуть или свернуть элемент, чтобы показать или скрыть его дочерние элементы;
- добавить к элементу дочерний элемент (со связанной картой или без нее);
- добавить к элементу несколько дочерних элементов (со связанными картами);
- редактировать элемент;
- удалить элемент (включая его дочерние элементы).



Настройка элемента

Чтобы настроить элемент дерева навигации, добавьте новый элемент или отредактируйте существующий элемент.



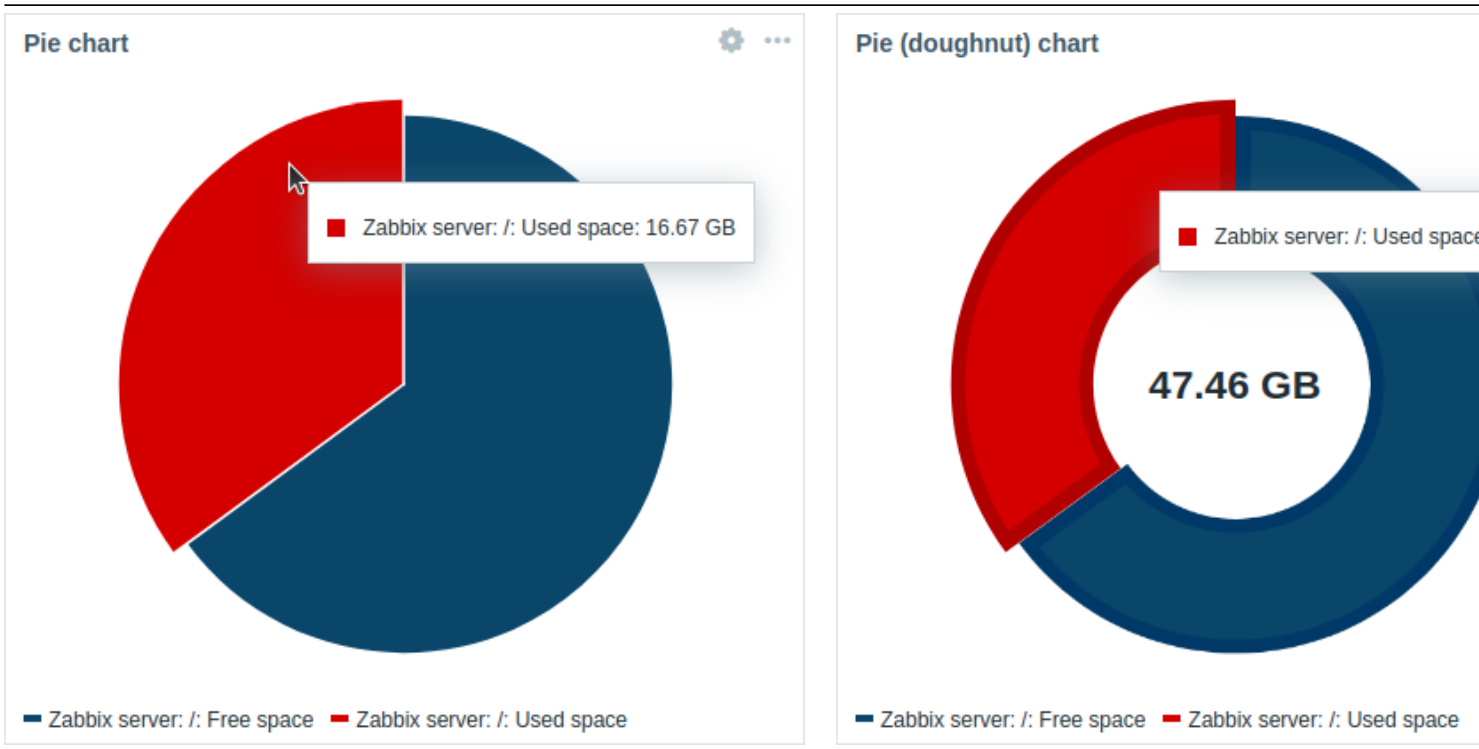
Доступны следующие параметры конфигурации элемента дерева навигации:

Имя (Name)	Введите имя элемента дерева навигации.
Связанная карта (Linked map)	Выберите карту, которая будет связана с элементом дерева навигации. Это поле с функцией автодополнения; таким образом, после начала ввода имени карты сети будет автоматически предложен список совпадающих карт в выпадающем меню.
Добавить подкарты (Add submaps)	Отметьте этот флажок, чтобы добавить к элементу дерева навигации подкарты связанной карты в качестве дочерних элементов.

Круговая диаграмма

Обзор

Виджет *Круговая диаграмма* отображает числовые данные элементов данных в виде векторной круговой или кольцевой диаграммы. Этот виджет помогает визуализировать вклад элементов данных или узлов сети в общий набор данных.

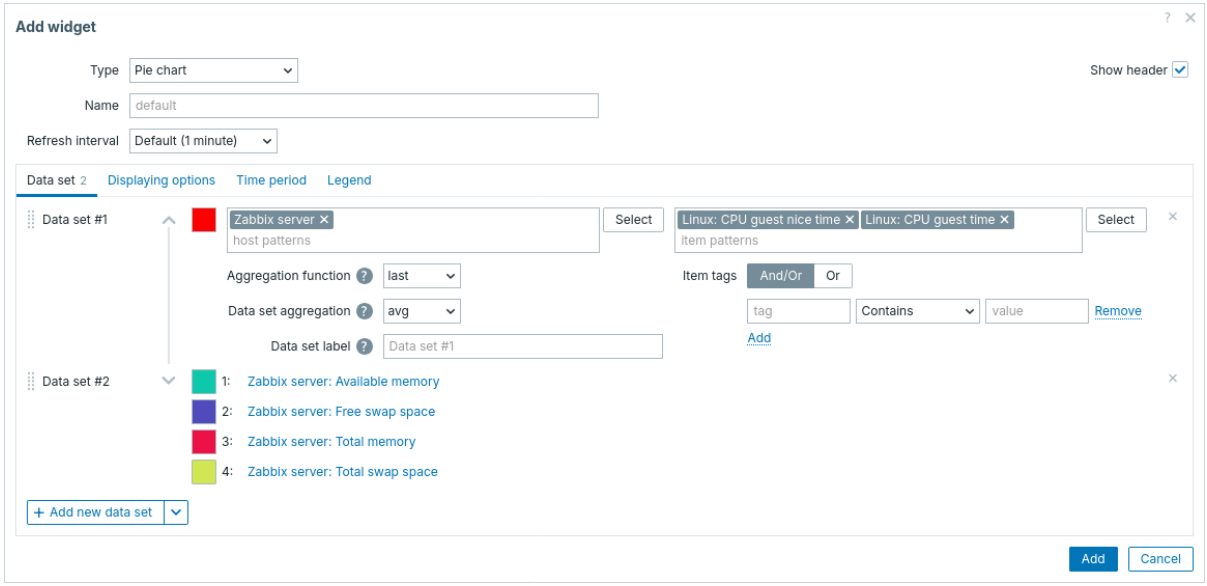


При наведении указателя мыши выделенный сектор расширяется и отображает свою легенду; щелчок по сектору оставляет его расширенным, а легенду видимой до тех пор, пока она не будет закрыта.

Информацию, отображаемую в виджете *Круговая диаграмма*, можно скачать в виде изображения PNG, выбрав пункт *Скачать изображение* в меню *Действия* виджета ..., которое появляется при наведении указателя мыши на виджет. В зависимости от настроек браузера изображение либо автоматически сохраняется в папке `Downloads`, либо открывается диалоговое окно *Сохранить*, позволяющее выбрать расположение файла.

Настройка

Для настройки выберите *Круговая диаграмма (Pie chart)* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбрать данные для круговой диаграммы путем добавления наборов данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* - отображаются данные соответствующих элементов данных. Можно выбрать один базовый цвет или строку палитры, чтобы назначить отдельные цвета каждому соответствующему элементу данных.
- *Список элементов данных* - отображаются данные выбранных элементов данных. Цвет каждого элемента данных можно выбрать отдельно с помощью палитры.

По умолчанию добавляется набор данных *Шаблоны элементов данных*.

Набор данных	Для набора данных **Шаблоны элементов данных**: Выберите или введите шаблоны узлов сети и элементов данных; на круговой диаграмме будут отображаться данные элементов данных, соответствующих введенным шаблонам; можно отобразить до 50 элементов данных. Для выбора можно использовать шаблоны с подстановочными знаками (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. Символ подстановки всегда интерпретируется, поэтому невозможно добавить, например, элемент данных с именем *item** отдельно, если существуют другие соответствующие элементы данных (например, *item2*, *item3*). Указание шаблонов узлов сети и элементов данных является обязательным для наборов данных ”Шаблоны элементов данных”. См. также: Подробные сведения о настройке набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для круговой диаграммы, нажав кнопку *Добавить элемент данных*. С помощью раскрывающегося списка *Тип* после имени элемента данных можно выбрать тип отображения для каждого элемента данных: **Обычный** - значение элемента данных отображается на круговой диаграмме пропорционально (по умолчанию); **Итоговый** - значение элемента данных занимает всю круговую диаграмму. Обратите внимание, что на одной круговой диаграмме может быть только один элемент данных типа ”Итоговый”; он будет размещен первым в легенде круговой диаграммы. Если для элемента данных задан тип ”Итоговый”, параметр *Агрегирование набора данных* (см. ниже) будет отключен и установлен в значение ”не используется”. Также можно выбрать совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*. Для виджетов применяются те же параметры, что и для отдельных элементов данных. Указание элементов данных или виджетов является обязательным для наборов данных ”Список элементов данных”. См. также: Подробные сведения о настройке набора данных. Обратите внимание, что разрешены только числовые типы элементов данных. При настройке виджета на панели шаблона параметр для указания шаблонов узлов сети недоступен, а параметр для указания списка элементов данных позволяет выбрать только элементы данных, настроенные в шаблоне.
Функция агрегирования	Укажите функцию агрегирования, которую следует использовать для каждого элемента данных в наборе данных в пределах выбранного *Периода времени*: **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **sum** - отображать сумму значений; **count** - отображать количество значений; **first** - отображать первое значение; **last** - отображать последнее значение (по умолчанию). Агрегирование позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день). Для *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для *count* нечисловые данные будут преобразованы в числовые.

Агрегирование набора данных	Укажите функцию агрегирования, которую следует использовать для всего набора данных: **не используется** - без агрегирования, элементы данных отображаются отдельно (по умолчанию); **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **sum** - отображать сумму значений; **count** - отображать количество значений. Агрегирование позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день), выбранный на вкладке *Период времени* или используемый для всей панели.
Метка набора данных	Укажите пользовательскую метку для набора данных. Метка отображается в настройках набора данных и легенде круговой диаграммы (для агрегированных наборов данных). Все наборы данных нумеруются, включая наборы с указанной *Меткой набора данных*. Если метка не указана, набор данных будет автоматически обозначен в соответствии с его номером (например, ”Набор данных №2”, ”Набор данных №3” и т. д.). Нумерация наборов данных пересчитывается после изменения порядка или перетаскивания наборов данных. Слишком длинные метки наборов данных будут сокращены, чтобы соответствовать месту отображения (например, ”Количество проц...”).
Теги элементов данных	Укажите теги для фильтрации элементов данных, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега будут объединены условием ИЛИ; **ИЛИ** - достаточно выполнения одного условия.

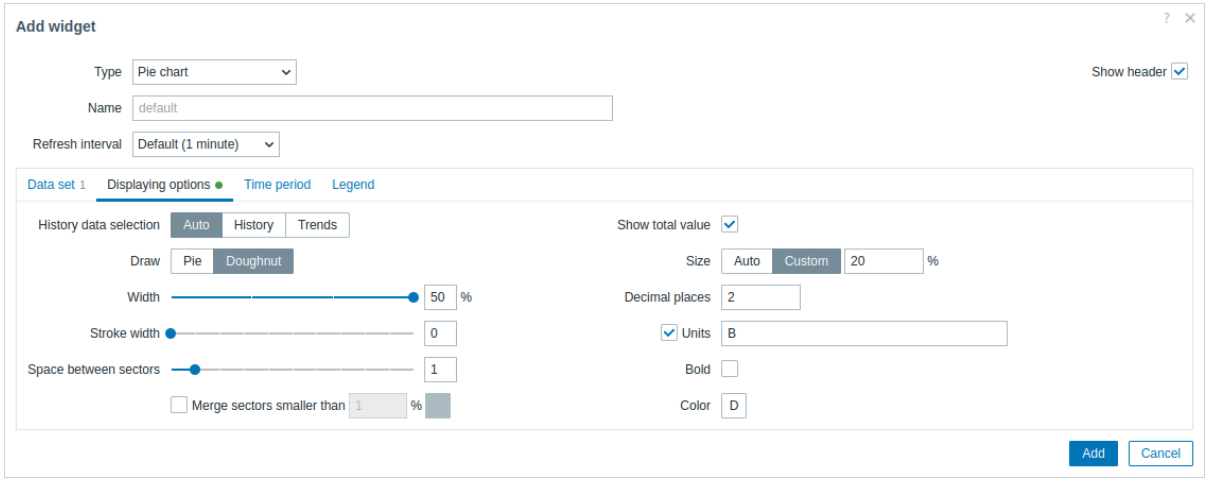
Подробности настройки набора данных

Существующие наборы данных отображаются в списке. Вы можете изменять их порядок, разворачивать/сворачивать, менять цвета и клонировать эти наборы данных.

Дополнительные сведения см. в разделе подробностей настройки набора данных в виджете *Graph*. Эти сведения также применимы к виджету *Pie chart*.

Параметры отображения

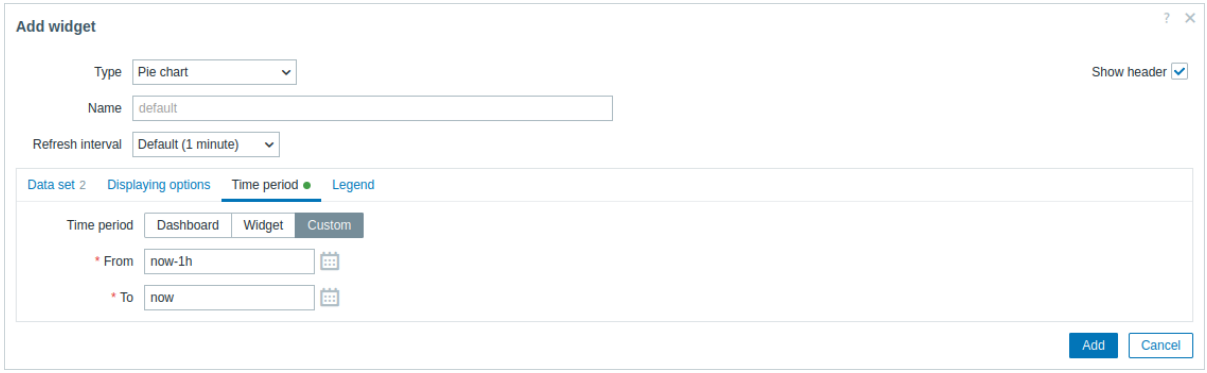
Вкладка **Параметры отображения** позволяет настроить выбор данных истории и параметры визуализации для круговой диаграммы:



Выбор данных истории	Выберите источник данных: **Авто** - данные выбираются в соответствии с классическим алгоритмом (по умолчанию); **История** - данные из истории; **Тренды** - данные из трендов.
Отображение	Выберите стиль визуализации круговой диаграммы: **Круговая** - полный круг (секторы занимают 100% радиуса); **Кольцевая** - круг с пустым пространством в центре (секторы занимают до 50% радиуса).
Расстояние между секторами	Выберите размер расстояния (в единицах от 0 до 10) между секторами (по умолчанию - «1»).
Объединять секторы меньше N%	Установите флажок, чтобы объединять секторы размером менее N%. Если параметр включен, выберите цвет объединенных секторов и пороговое значение в процентах (N) для объединения малых секторов.
Стиль отображения:	
Кольцевая	
Ширина	Выберите ширину кольца: 20, 30, 40 или 50% (по умолчанию) радиуса.
Ширина границы	Выберите ширину границы сектора кольца (0-10).
Показывать общее значение	Установите флажок, чтобы отображать общее значение в центре кольцевой диаграммы.
Размер	Выберите параметр размера общего значения: **Авто** - размер текста автоматически подбирается так, чтобы он хорошо читался в центре кольца; **Пользовательский** - укажите размер текста в процентах от общей высоты виджета.
Знаков после запятой	Укажите количество знаков после запятой для общего значения (0-6).
Единицы измерения	Укажите единицы измерения для общего значения.
Полужирный	Установите флажок, чтобы отображать общее значение полужирным шрифтом.
Цвет	Выберите цвет общего значения.

Период времени

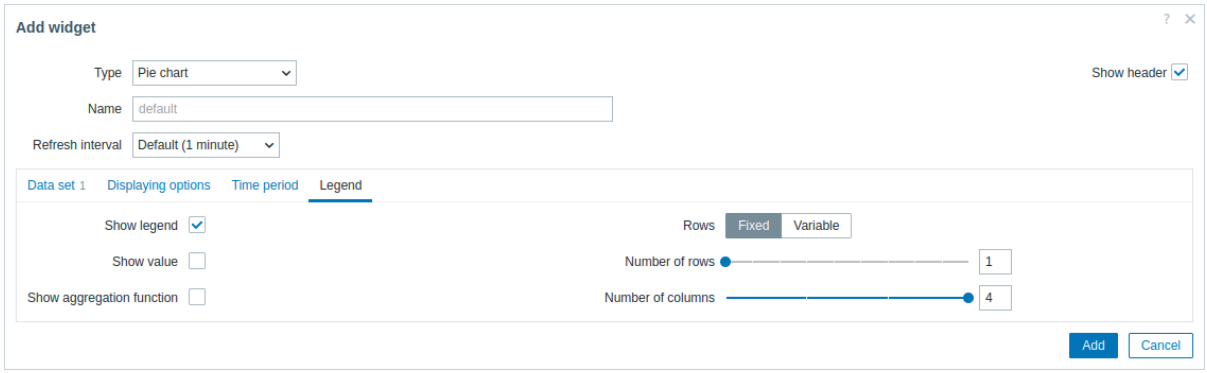
Вкладка **Период времени** позволяет задать пользовательский период времени для параметров агрегации круговой диаграммы:



Период времени	Выберите источник данных для периода времени: **Панель мониторинга** — использовать селектор периода времени панели мониторинга; **Виджет** — использовать совместимый виджет (указанный в параметре *Виджет*); **Пользовательский** — использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении указателя мыши. Обратите внимание, что независимо от настройки *Период времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.
Виджет	Введите или выберите совместимый виджет (*График*, *График (классический)*, *Прототип графика*) в качестве источника данных для периода времени. Этот параметр доступен, если для параметра *Период времени* установлено значение «Виджет».
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».

Легенда

Вкладка **Легенда** позволяет настроить легенду круговой диаграммы:



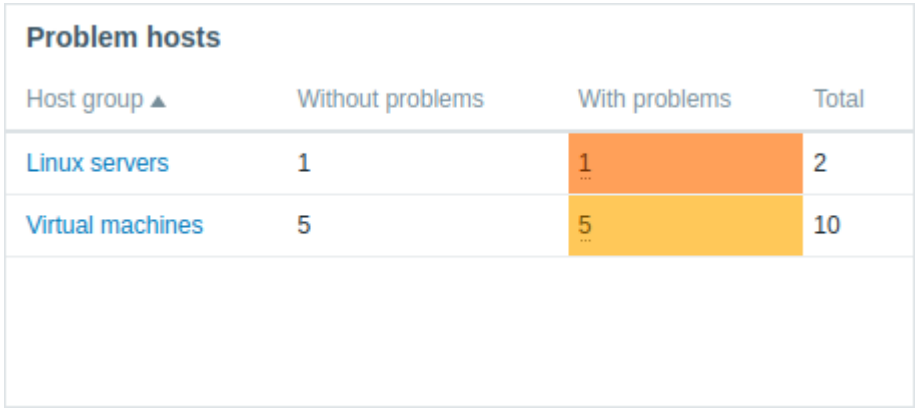
Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на круговой диаграмме (по умолчанию флажок установлен).
Показывать значение	Установите этот флажок, чтобы показывать значение элемента данных в легенде.
Показывать функцию агрегации	Установите этот флажок, чтобы показывать функцию агрегации в легенде.
Строки	Выберите режим отображения строк легенды: **Фиксированное количество** - количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное количество** - количество отображаемых строк определяется количеством настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.

Количество строк/ Максимальное количество строк	Если для параметра *Строки* выбрано значение «Фиксированное количество», укажите количество отображаемых строк легенды (1-10). Если для параметра *Строки* выбрано значение «Переменное количество», укажите максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Укажите количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Показывать значение* не установлен.

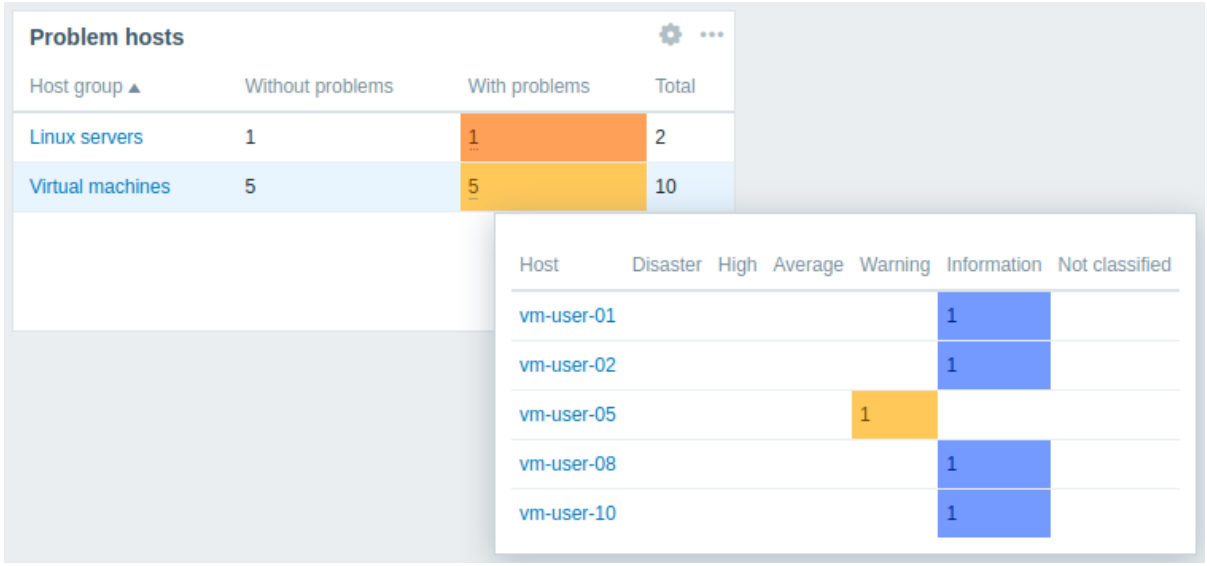
Проблемные узлы сети

Обзор

Виджет *Проблемные узлы сети* отображает количество проблем-источников для каждой группы узлов сети и показывает наивысшую степень серьезности проблемы в каждой группе. Он позволяет фильтровать проблемы по различным параметрам (группы узлов сети, узлы сети, названия проблем и т. д.).

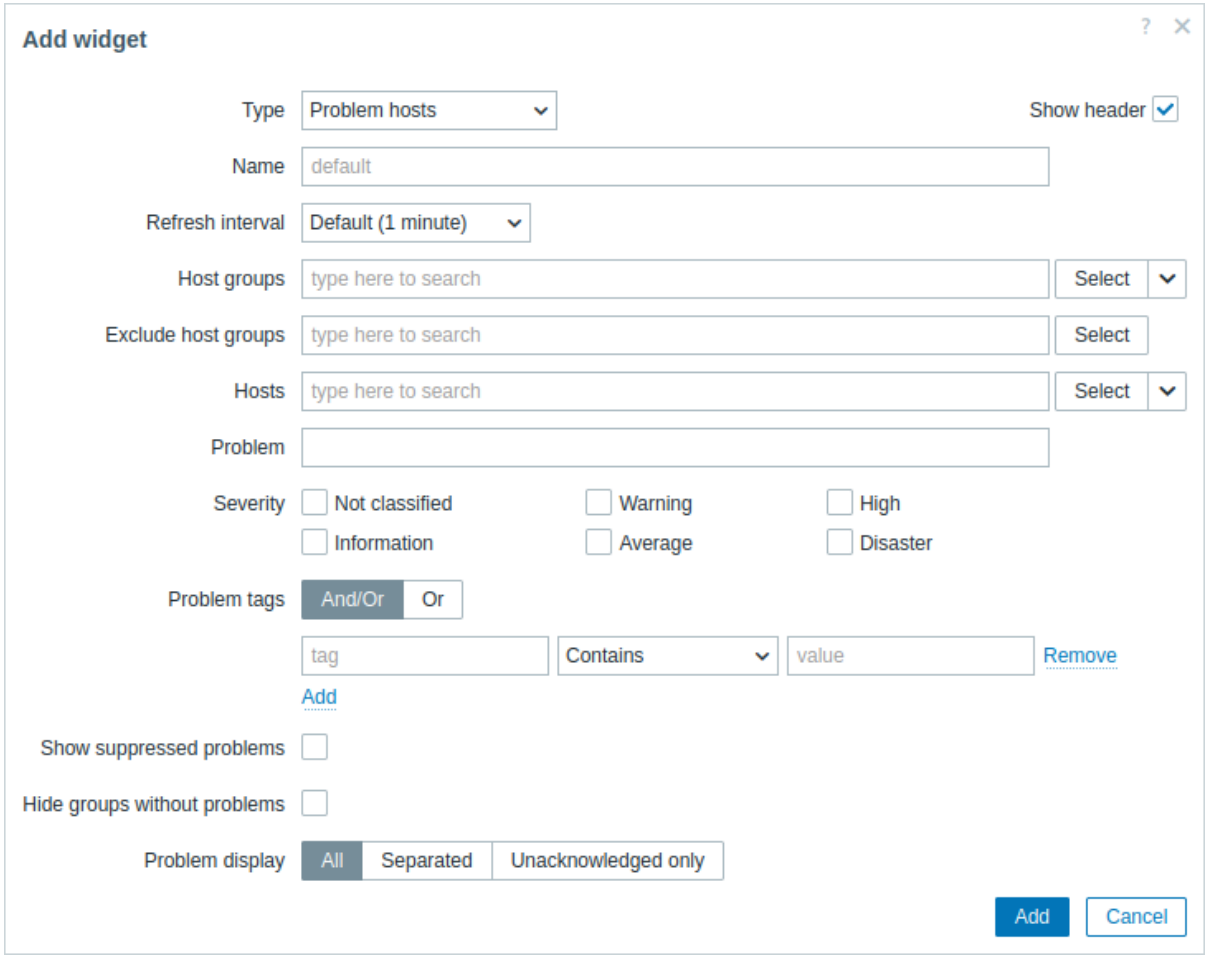


При наведении на количество проблем отображается список узлов сети с проблемами:



Настройка

Для настройки выберите *Узлы сети с проблемами (Problem hosts)* в качестве типа:



В дополнение к параметрам, которые являются общими для всех виджетов, вы можете установить следующие специфичные опции:

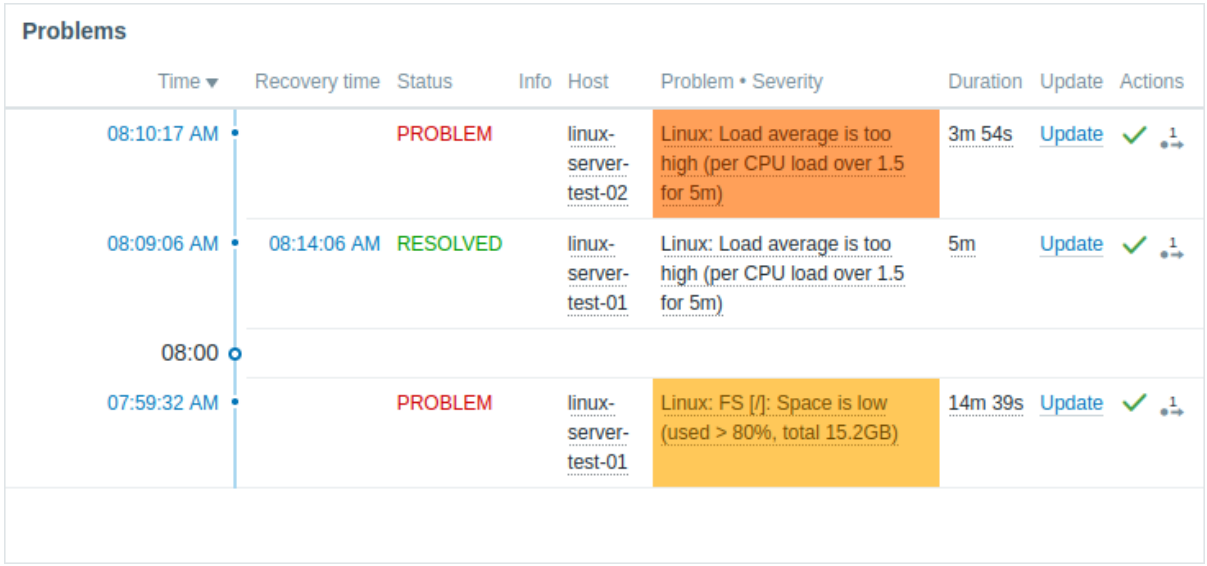
Группы узлов сети (Host groups)	Выберите группы узлов сети для отображения в виджете. В качестве альтернативы выберите для групп узлов сети в качестве источника данных совместимый виджет. Это поле с функцией автодополнения; таким образом, после начала ввода имени группы будет автоматически предложен список совпадающих групп в выпадающем меню. При выборе родительской группы узлов сети неявно выбираются все вложенные группы узлов сети. Данные от этих групп узлов сети будут отображаться в виджете; если не выбрана ни одна группа, то будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Исключить группы узлов сети (Exclude host groups)	Выберите группы узлов сети, которые нужно скрыть от виджета. Это поле с функцией автодополнения; таким образом, после начала ввода имени группы будет автоматически предложен список совпадающих групп в выпадающем меню. При выборе родительской группы узлов сети неявно выбираются все вложенные группы узлов сети. Данные узлов сети из этих групп не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут быть в группе «Group A», а узлы сети 002, 003 — также и в группе «Group B». Если мы одновременно выбираем *показать* группу «Group A» и *исключить* группу «Group B», то на панели будут отображены данные только с узла сети 001. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети (Hosts)	Выберите узлы сети для отображения в виджете. В качестве альтернативы выберите для узлов сети в качестве источника данных совместимый виджет или панель. Это поле с функцией автодополнения; таким образом, после начала ввода имени узла сети будет автоматически предложен список совпадающих узлов сети в выпадающем меню. Если не введено никаких узлов сети, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Проблема (Problem)	Количество отображаемых узлов сети с проблемами можно ограничить по имени проблемы. Если ввести здесь строку, будут отображаться только те узлы сети, чьи имена проблем содержат введённую строку. Макросы не раскрываются.
Важность (Severity)	Укажите важности проблем, чтобы отфильтровать проблемы для отображения в виджете. Если никакие важности не отмечены, будут отображаться все проблемы.
Теги проблем (Problem tags)	Укажите теги проблем для ограничения количества проблем, отображаемых в виджете. Можно как включать, так и исключать определённые теги и значения тегов. Можно указать несколько условий. Поиск соответствия имён тегов всегда чувствителен к регистру. По каждому условию доступно несколько операторов: **Существует (Exists)** — включить указанные имена тегов; **Равно (Equals)** — включить указанные имена тегов и их значения (с учётом регистра); **Содержит (Contains)** — включить указанные имена тегов, в которых значения тегов содержат введённую строку (совпадение подстроки, без учёта регистра); **Не существует (Does not exist)** — исключить указанные имена тегов; **Не равно (Does not equal)** — исключить указанные имена тегов и их значения (с учётом регистра); **Не содержит (Does not contain)** — исключить имена указанных тегов, в которых значения тегов содержат введённую строку (совпадение подстроки, без учёта регистра). Для условий имеется два типа вычислений: **И/Или (And/Or)** — все условия должны соблюдаться, условия с одинаковыми именами тегов будут сгруппированы с условием *Или (Or)*; **Или (Or)** — достаточно соблюдения одного условия.
Подавленные проблемы (Show suppressed problems)	Установите флажок для отображения проблем, которые в противном случае могут быть подавлены (не показаны) по причине режима обслуживания узла сети.
Скрыть группы без проблем (Hide groups without problems)	Отметьте опцию *Скрыть группы без проблем (Hide groups without problems)*, чтобы скрыть в виджете данные от групп узлов сети, в которых проблемы отсутствуют. Этот параметр недоступен при настройке виджета на панели шаблона.
Отображение проблем (Problem display)	Отображать количество проблем как: **Все (All)** — будет отображаться общее количество проблем; **Раздельно (Separated)** — количество неподтвержденных проблем будет отображаться отдельным числом от общего количества проблем; **Только неподтверждённые (Unacknowledged only)** — будет отображаться только количество неподтверждённых проблем.

Проблемы

Обзор

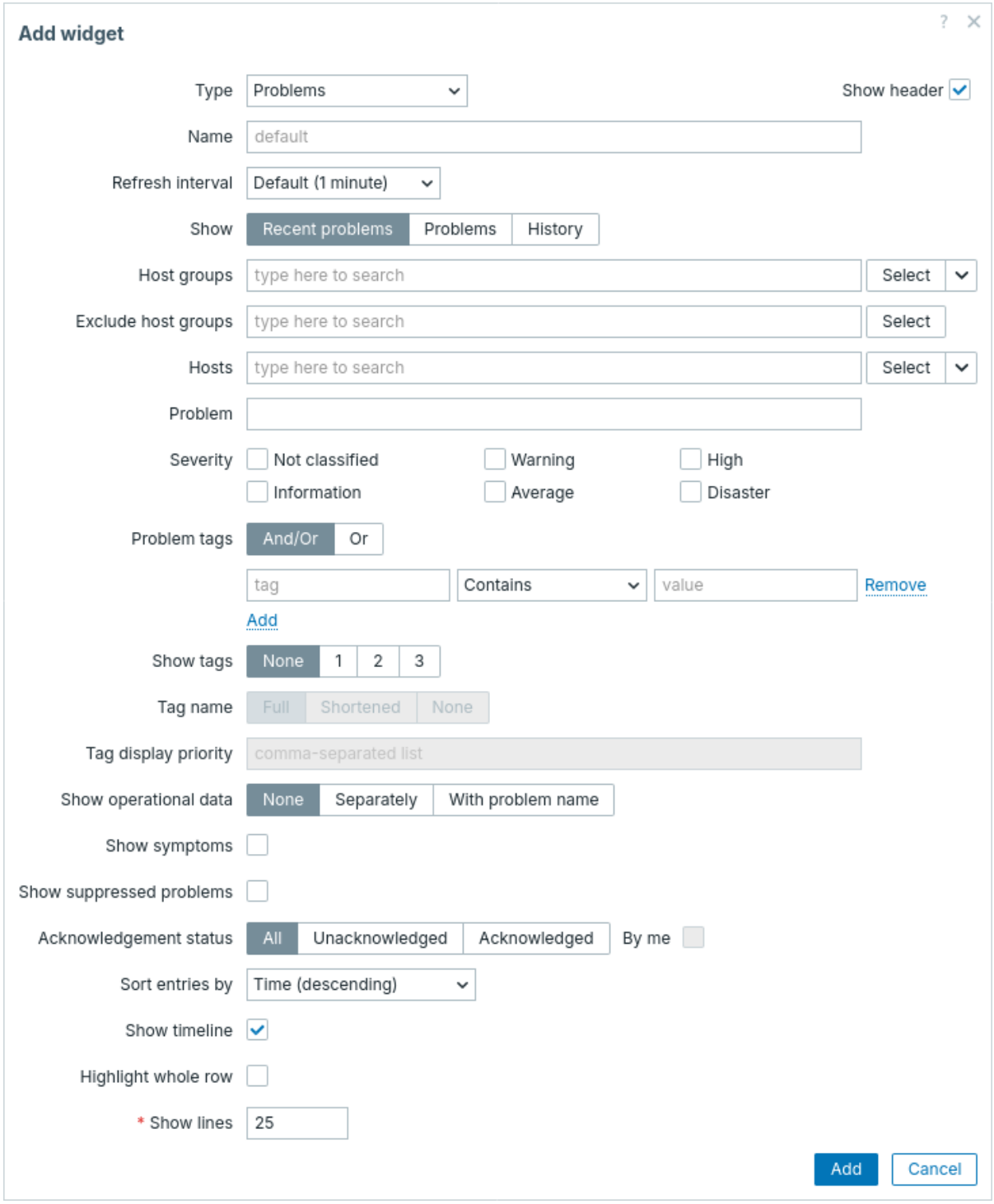
Виджет *Problems* отображает текущие проблемы, отфильтрованные по различным параметрам (группы узлов сети, узлы сети, названия проблем и т. д.), предоставляя вам наглядное представление о том, что требует внимания.



Он отображает те же данные, что и *Monitoring* > *Problems*, и может показывать до 1000 записей.

Конфигурация

Чтобы настроить, выберите *Problems* в качестве типа:



Вы можете ограничить количество проблем, отображаемых в виджете, различными способами — по статусу проблемы, имени проблемы, серьезности, группе узлов сети, узлу сети, тегу события, статусу подтверждения и т. д.

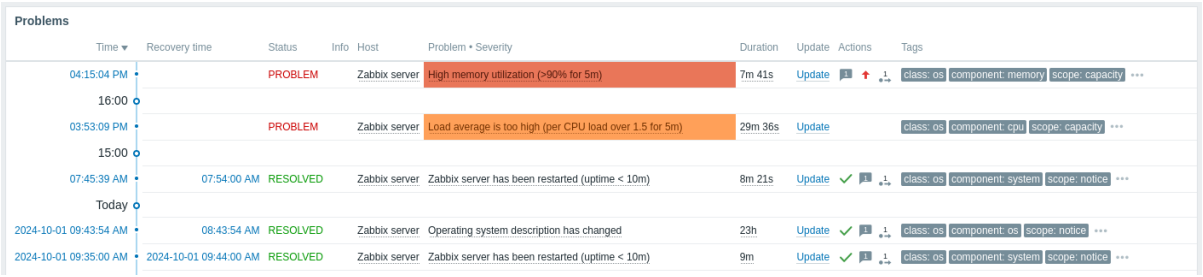
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Show	Фильтрация по статусу проблемы: **Recent problems** - отображаются нерешенные и недавно решенные проблемы (по умолчанию); **Problems** - отображаются нерешенные проблемы; **History** - отображается история всех событий.

Host groups	Выберите группы узлов сети, проблемы которых нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Проблемы из этих групп узлов сети будут отображаться в виджете; если группы узлов сети не указаны, будут отображаться проблемы из всех групп узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Exclude host groups	Выберите группы узлов сети, проблемы которых нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Проблемы из этих групп узлов сети не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут входить в Group A, а узлы сети 002, 003 — также в Group B. Если одновременно выбрать *show* для Group A и *exclude* для Group B, в виджете будут отображаться только проблемы узла сети 001. Этот параметр недоступен при настройке виджета на шаблонной панели.
Hosts	Выберите узлы сети, проблемы которых нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться проблемы всех узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Problem	Можно ограничить количество отображаемых проблем по их имени. Если ввести здесь строку, будут отображаться только те проблемы, в имени которых содержится введенная строка. Макросы не раскрываются.
Severity	Отметьте серьезности проблем, чтобы отфильтровать проблемы, отображаемые в виджете. Если ни одна серьезность не отмечена, будут отображаться все проблемы.
Problem tags	Укажите теги проблемы, чтобы ограничить количество проблем, отображаемых в виджете. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (сопоставление подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (сопоставление подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия. При фильтрации указанные здесь теги будут отображаться первыми вместе с проблемой, если это не переопределено списком *Tag display priority* (см. ниже).
Show tags	Выберите количество отображаемых тегов: **None** - столбец *Tags* отсутствует; **1** - столбец *Tags* содержит один тег; **2** - столбец *Tags* содержит два тега; **3** - столбец *Tags* содержит три тега. Чтобы увидеть все теги проблемы, наведите указатель мыши на значок с тремя точками.

Tag name	Выберите режим отображения имен тегов: **Full** - имена и значения тегов отображаются полностью; **Shortened** - имена тегов сокращаются до 3 символов, но значения тегов отображаются полностью; **None** - отображаются только значения тегов; имена не отображаются.
Tag display priority	Введите приоритет отображения тегов для проблемы в виде списка тегов, разделенного запятыми. Следует использовать только имена тегов, без значений. Пример: `Services,Applications,Application` Теги из этого списка всегда будут отображаться первыми, переопределяя естественный алфавитный порядок.
Show operational data	Выберите режим отображения оперативных данных: **None** - оперативные данные не отображаются; **Separately** - оперативные данные отображаются в отдельном столбце; **With problem name** - добавлять оперативные данные к имени проблемы, заключая оперативные данные в скобки. Подробности см. в описании поля *Operational data* в конфигурации триггера.
Show symptoms	Отметьте флажок, чтобы отображать проблемы, классифицированные как симптомы, в отдельной строке.
Show suppressed problems	Отметьте флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети или отдельного подавления проблем.
Acknowledgement status	Фильтр для отображения всех проблем, только неподтвержденных проблем или только подтвержденных проблем. Отметьте дополнительный флажок, чтобы исключить проблемы, которые когда-либо были подтверждены вами.
Sort entries by	Сортировать записи по: **Time** (по убыванию или возрастанию); **Severity** (по убыванию или возрастанию); **Problem name** (по убыванию или возрастанию); **Host** (по убыванию или возрастанию). Сортировка записей по **Host** (по убыванию или возрастанию) недоступна при настройке виджета на шаблонной панели.
Show timeline	Отметьте флажок, чтобы отображать визуальную временную шкалу.
Highlight whole row	Отметьте флажок, чтобы выделять всю строку для нерешенных проблем. Для выделения используется цвет серьезности проблемы. *Highlight whole row* недоступно в темах с высокой контрастностью.
Show lines	Укажите количество строк проблем для отображения.

Использование виджета



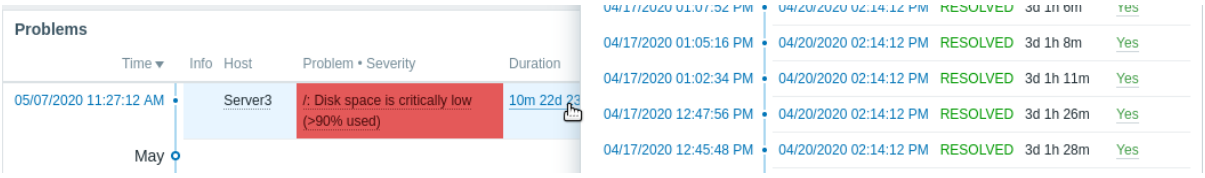
Виджет проблем предоставляет быстрый доступ к дополнительной информации:

- Щелкните по дате проблемы (в столбце Time) или дате восстановления (в столбце Recovery Time), чтобы просмотреть сведения о событии.
- Если поле Info не пустое, наведите указатель на отображаемый значок, чтобы просмотреть дополнительные сведения.
- Щелкните по имени узла сети, чтобы открыть меню узла сети.
- Щелкните по имени проблемы, чтобы открыть меню события.
- Наведите указатель на длительность проблемы или щелкните по ней, чтобы просмотреть всплывающее окно события проблемы.
- Щелчок по *Update* открывает окно Обновление проблемы.
- Наведите указатель на серый значок стрелки в столбце Actions или щелкните по нему, чтобы просмотреть список

выполненных действий.

Всплывающее окно события проблемы

Всплывающее окно события проблемы содержит список событий проблемы для этого триггера и, если задано, описание триггера и кликабельный URL.



Чтобы открыть всплывающее окно события проблемы:

- Наведите указатель мыши на длительность проблемы в столбце *Duration* виджета *Problems*. Всплывающее окно исчезает, как только вы убираете указатель мыши с длительности.
- Щелкните по длительности в столбце *Duration* виджета *Problems*. Всплывающее окно исчезает только если вы щелкнете по длительности еще раз.

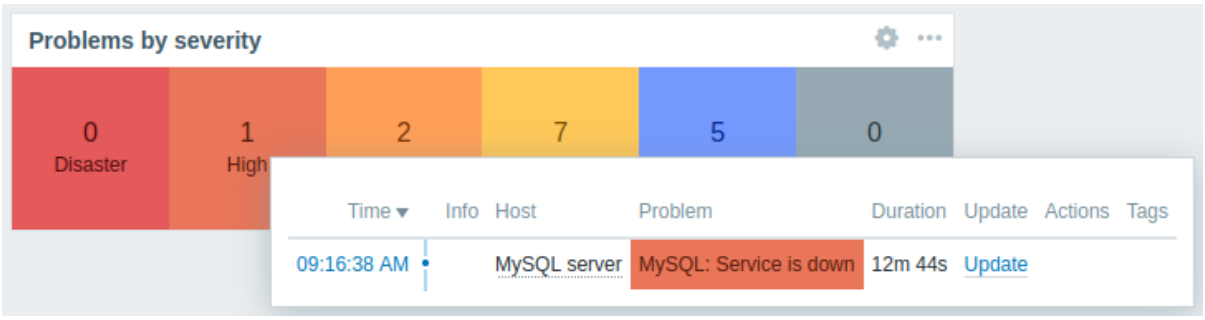
Проблемы по степени серьезности

Обзор

Виджет *Проблемы по степени важности* отображает количество причинных проблем, сгруппированных по степени важности. Он позволяет фильтровать проблемы по различным параметрам (группы узлов сети, узлы сети, названия проблем и т. д.).

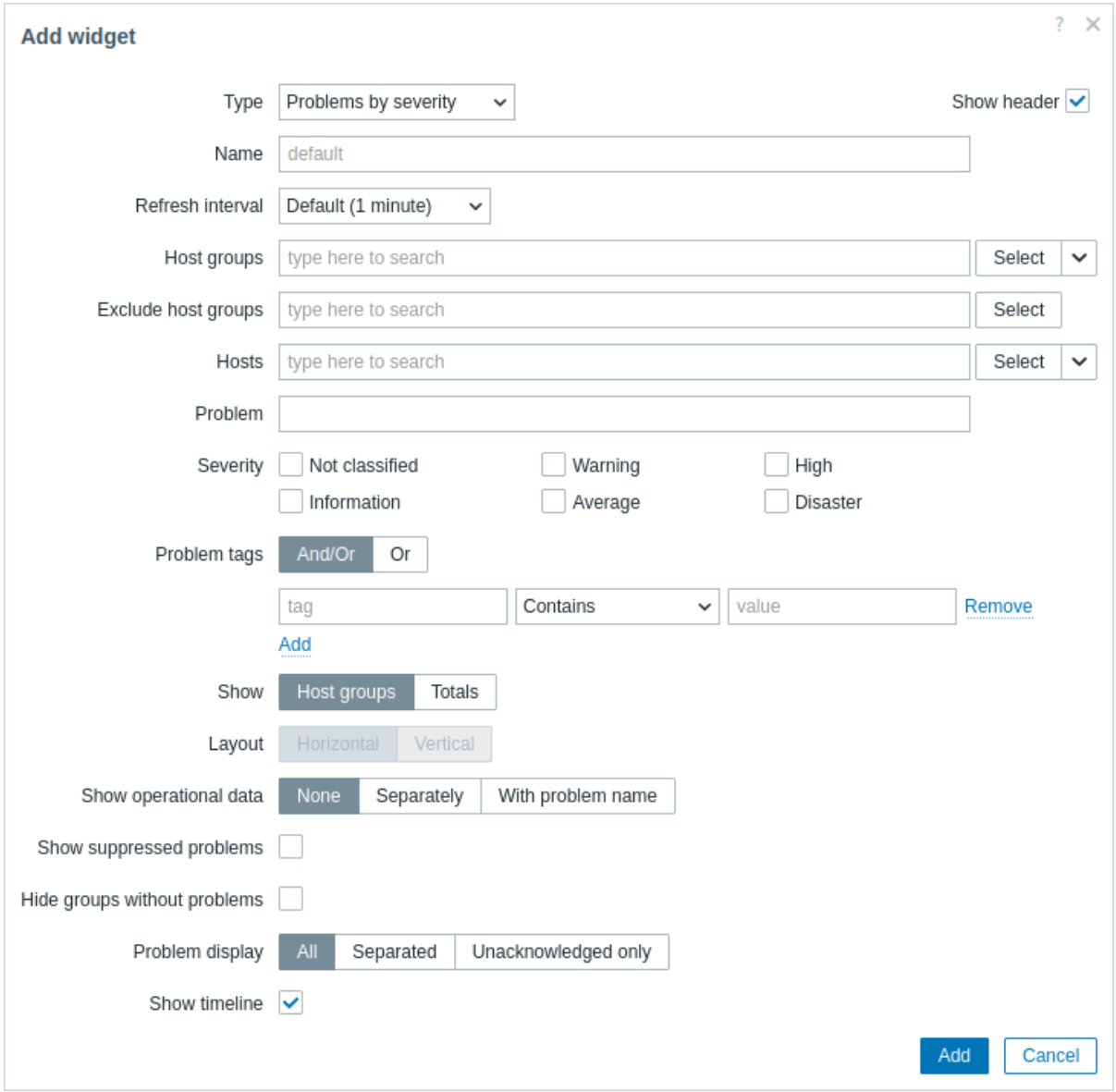


При наведении курсора на количество проблем отображается список связанных проблем:



Конфигурация

Чтобы настроить, выберите *Problems by severity* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

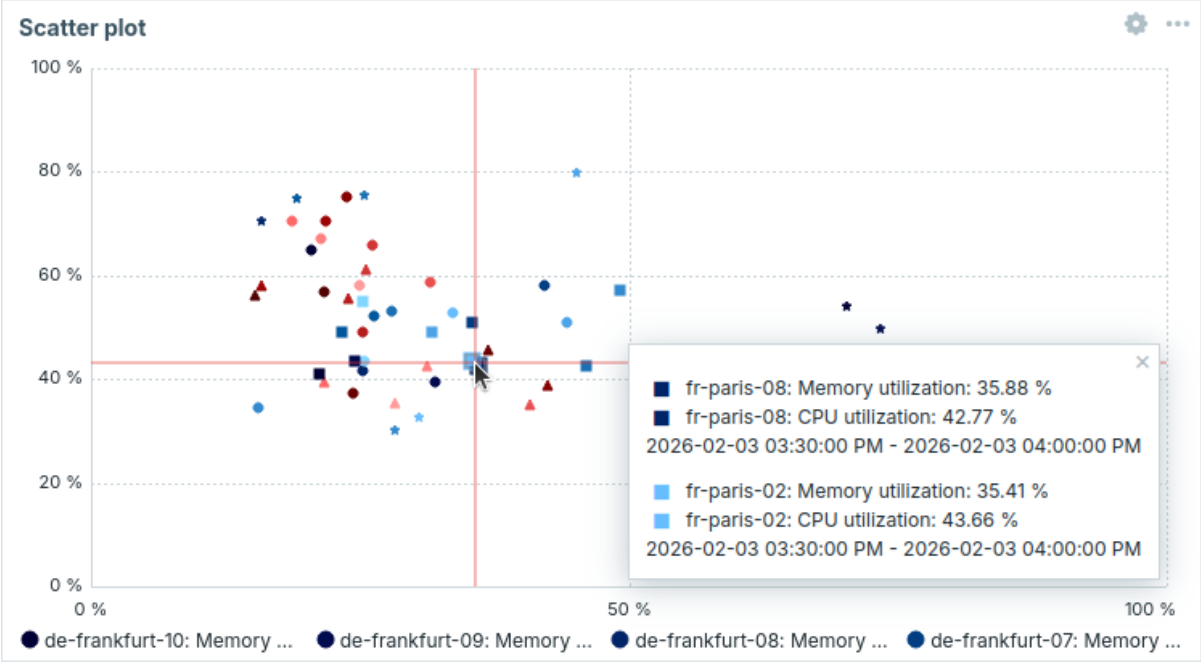
Host groups	Выберите группы узлов сети, которые нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп будут отображаться в виджете; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Exclude host groups	Выберите группы узлов сети, которые нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут входить в Group A, а узлы сети 002, 003 — также в Group B. Если одновременно выбрать *show* Group A и *exclude* Group B, на панели будут отображаться только данные узла сети 001. Этот параметр недоступен при настройке виджета на шаблонной панели.

Hosts	Выберите узлы сети, которые нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Problem	Можно ограничить число отображаемых узлов сети с проблемами по имени проблемы. Если здесь ввести строку, будут отображаться только те узлы сети, у которых есть проблемы, содержащие введенную строку в имени. Макросы не раскрываются.
Severity	Отметьте уровни серьезности проблем, чтобы отфильтровать проблемы, отображаемые в виджете. Если ни один уровень серьезности не отмечен, будут отображаться все проблемы.
Problem tags	Укажите теги проблем, чтобы ограничить число проблем, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия.
Show	Выберите параметр отображения: **Host groups** - отображать проблемы по группам узлов сети; **Totals** - отображать общее число проблем для всех выбранных групп узлов сети в цветных блоках, соответствующих уровню серьезности проблемы. Этот параметр недоступен при настройке виджета на шаблонной панели, и будет отображаться только общее число проблем.
Layout	Выберите вариант расположения: **Horizontal** - цветные блоки с итогами будут отображаться горизонтально; **Vertical** - цветные блоки с итогами будут отображаться вертикально. Этот параметр доступен, если для *Show* установлено значение ”Totals”.
Show operational data	Отображать оперативные данные как: **None** - оперативные данные не будут отображаться **Separately** - оперативные данные будут отображаться отдельной строкой **With problem name** - оперативные данные будут отображаться вместе с именем проблемы. Подробности см. в описании поля *Operational data* в настройке триггера.
Show suppressed problems	Отметьте этот флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
Hide groups without problems	Отметьте параметр *Hide groups without problems*, чтобы скрыть в виджете данные из групп узлов сети без проблем. Этот параметр недоступен при настройке виджета на шаблонной панели.
Problem display	Отображать число проблем как: **All** - будет отображаться полное число проблем; **Separated** - число неподтвержденных проблем будет отображаться отдельно от общего числа проблем; **Unacknowledged only** - будет отображаться только число неподтвержденных проблем.
Show timeline	Отметьте флажок, чтобы отображать визуальную временную шкалу.

Диаграмма рассеяния

Обзор

Виджет *Точечная диаграмма* отображает взаимосвязь между двумя метриками, нанося отдельные точки данных на оси X и Y. Это помогает выявить закономерности, кластеры, корреляции и выбросы в наборе данных.



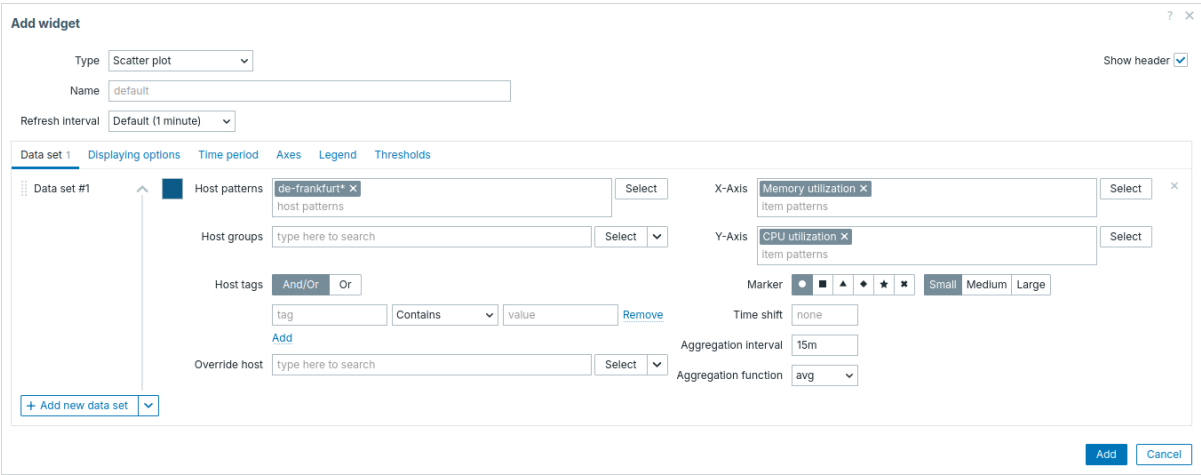
При наведении указателя на точку данных отображается всплывающая подсказка со списком элементов данных из этой точки и других точек в радиусе 6 пикселей; сначала список сортируется по значениям оси X точечной диаграммы в порядке возрастания, а затем по значениям оси Y в порядке убывания.

Выбор элемента данных во всплывающей подсказке передает его данные другим виджетам.

Информацию, отображаемую в виджете *Точечная диаграмма*, можно скачать в виде изображения PNG, выбрав пункт *Скачать изображение* в меню *Действия* виджета ..., которое появляется при наведении указателя на виджет. В зависимости от настроек браузера изображение либо автоматически сохраняется в папке `Downloads`, либо открывается диалоговое окно *Сохранить*, в котором можно выбрать расположение файла.

Настройка

Чтобы настроить, выберите *Scatter plot* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбирать данные для точечной диаграммы путем добавления наборов данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* - узлы сети с соответствующими элементами данных отображаются как точки данных. Можно выбрать один базовый цвет или строку палитры, чтобы назначить разные цвета каждому узлу сети с соответствующими элементами данных.

- *Список элементов данных* - выбранные элементы данных отображаются как точки данных, где каждая точка представляет собой пару элементов данных (X и Y). Цвет каждого набора данных можно выбрать отдельно с помощью палитры.

Точки данных создаются только для тех пар узлов сети и элементов данных, у которых есть хотя бы одно значение по обеим осям.

По умолчанию добавляется набор данных *Шаблоны элементов данных*.

Набор данных	Для набора данных **Шаблоны элементов данных**: Выберите или введите шаблоны узлов сети и элементов данных; данные элементов данных, соответствующих этим шаблонам, будут отображаться на точечной диаграмме. Обратите внимание: если количество точек данных превышает 8 тыс., следует увеличить `memory_limit` в файле конфигурации `php.ini`. Рекомендуемый лимит памяти для 20 тыс. точек данных составляет 512 МБ. Для выбора можно использовать шаблоны с подстановочными символами (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным символом, введите строку вручную и нажмите *Enter*. Символ подстановки всегда интерпретируется, поэтому невозможно добавить, например, элемент данных с именем *item** отдельно, если существуют другие соответствующие элементы данных (например, *item2*, *item3*). Указание шаблонов узлов сети и элементов данных является обязательным для наборов данных «Шаблоны элементов данных». См. также: Подробные сведения о настройке набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для точечной диаграммы, нажав кнопку *Добавить элемент данных*. Также можно выбрать совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*. Указание элементов данных или виджетов является обязательным для наборов данных «Список элементов данных». См. также: Подробные сведения о настройке набора данных. Обратите внимание, что разрешены только числовые типы элементов данных. При настройке виджета на панели шаблона параметр для указания шаблонов узлов сети недоступен, а параметр для указания списка элементов данных позволяет выбирать только элементы данных, настроенные в шаблоне.
Группы узлов сети	Выберите группы узлов сети. Также можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение: после начала ввода имени группы появится раскрывающийся список соответствующих групп. Указание родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп будут отображаться в виджете; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр доступен только для наборов данных *Шаблоны элементов данных* и недоступен при настройке виджета на панели шаблона.

Теги узлов сети	Укажите теги для фильтрации узлов сети, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра); **Не существует** - исключить указанные имена тегов; **Не равно** - исключить указанные имена тегов и значения (с учетом регистра); **Не содержит** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки без учета регистра). Для условий доступны два типа вычисления: **И/ИЛИ** - должны выполняться все условия; условия с одинаковым именем тега будут объединены оператором «ИЛИ»; **ИЛИ** - достаточно выполнения одного условия. Этот параметр доступен только для наборов данных *Шаблоны элементов данных* и недоступен при настройке виджета на панели шаблона.
Переопределить узел сети	Выберите совместимый виджет или селектор узлов сети панели в качестве источника данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Маркер	Выберите тип и размер маркера для точек данных узлов сети в наборе данных.
Смещение времени	При необходимости укажите смещение времени. В этом поле можно использовать суффиксы времени. Допускаются отрицательные значения.
Функция агрегации	Укажите функцию агрегации, которая будет использоваться для каждого элемента данных в наборе данных в пределах выбранного *Интервала агрегации*: **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **sum** - отображать сумму значений; **count** - отображать количество значений (если по обеим осям существует хотя бы одно значение); **first** - отображать первое значение; **last** - отображать последнее значение. Агрегация позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день) вместо всех значений. Для *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для *count* нечисловые данные будут преобразованы в числовые.
Интервал агрегации	Укажите интервал для агрегации значений. В этом поле можно использовать суффиксы времени. Числовое значение без суффикса будет интерпретироваться как секунды. Обратите внимание: если виджет настроен на отображение исторических данных на основе трендов (*Выбор данных истории* установлен в *Тренды* или *Авто*), рекомендуется использовать интервал агрегации, кратный 1 часу (например, 3600, 60m, 1h, 3h и т. д.). Тренды хранят агрегированные значения с почасовой детализацией, поэтому использование интервала агрегации, не кратного 1 часу (например, 100s, 7min, 15min, 90min и т. д.), может привести к результатам, которые сложно интерпретировать. Указание интервала агрегации является обязательным. Интервал агрегации определяет временной интервал и координаты точки данных; например, если элементы данных по оси X имеют 4 значения, а элементы данных по оси Y - 1 значение, координаты отражают агрегацию 4 значений по X и 1 значения по Y в указанном интервале.

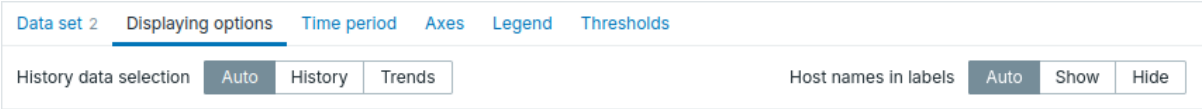
Подробности настройки набора данных

Существующие наборы данных отображаются в виде списка. Вы можете изменять их порядок, разворачивать/сворачивать, менять цвета и клонировать эти наборы данных.

Дополнительную информацию см. в разделе о подробностях настройки набора данных в виджете *Graph*. Эти сведения также применимы к виджету *Scatter plot*.

Параметры отображения

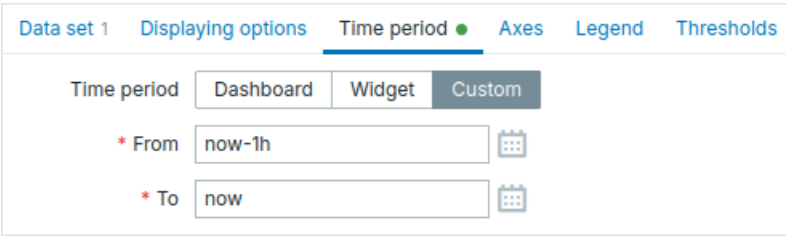
Вкладка **Параметры отображения** позволяет определить выбор данных истории:



Выбор данных истории	Укажите источник данных для точечной диаграммы: **Авто** — данные выбираются в соответствии с классическим алгоритмом построения графика (по умолчанию). **История** — данные истории. **Тренды** — данные трендов.
Имена узлов сети в подписях	Выберите, отображать ли имена узлов сети в легенде графика и во всплывающей подсказке, которая появляется при наведении указателя на данные графика: **Авто** — имена узлов сети отображаются, только если в наборах данных присутствует более одного узла сети (по умолчанию). **Показать** — имена узлов сети отображаются. **Скрыть** — имена узлов сети скрыты. Этот параметр недоступен при настройке виджета на панели мониторинга шаблона.

Период времени

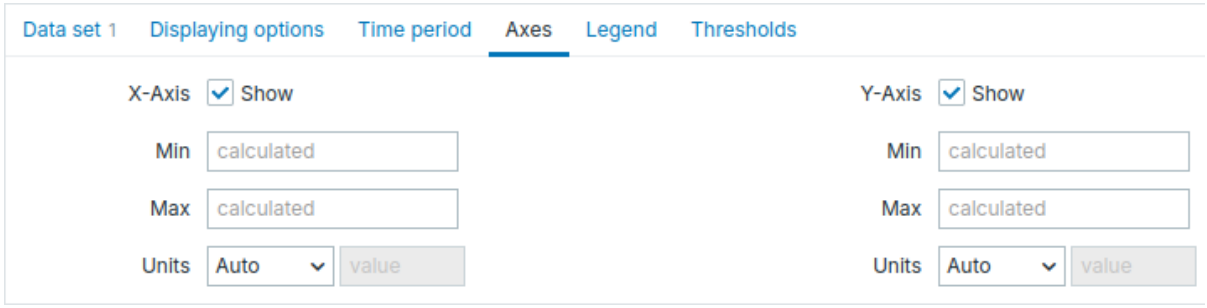
Вкладка **Период времени** позволяет задать период времени, за который будут отображаться данные на точечной диаграмме:



Период времени	Выберите источник данных для периода времени: **Панель мониторинга** — использовать селектор периода времени панели мониторинга; **Виджет** — использовать совместимый виджет (указанный в параметре *Виджет*); **Пользовательский** — использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, при наведении указателя мыши показывающий заданное время. Обратите внимание, что независимо от настройки *Период времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для параметра *Период времени* установлено значение «Виджет».
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для параметра *Период времени* установлено значение «Пользовательский».

Оси

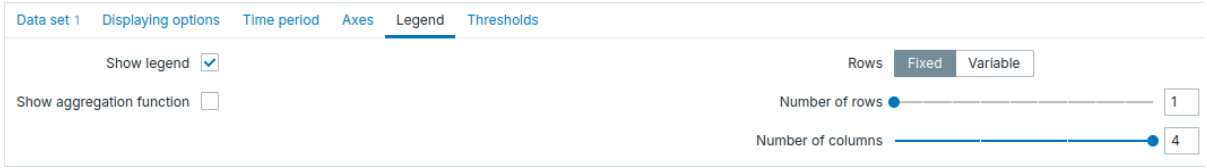
Вкладка **Оси** позволяет настроить отображение осей:



Ось X/Ось Y	Снимите этот флажок, чтобы скрыть ось X/ось Y (по умолчанию флажок установлен).
Минимум	Задайте минимальное значение соответствующей оси.
Максимум	Задайте максимальное значение соответствующей оси.
Единицы измерения	Выберите единицу измерения для значений оси точечной диаграммы из раскрывающегося списка: **Авто** — значения оси отображаются с использованием единицы измерения первого элемента данных в наборе данных; **Статические** — значения оси отображаются с использованием единицы измерения, указанной в поле ввода *значение*; если поле оставить пустым, отображаются только числовые значения.

Легенда

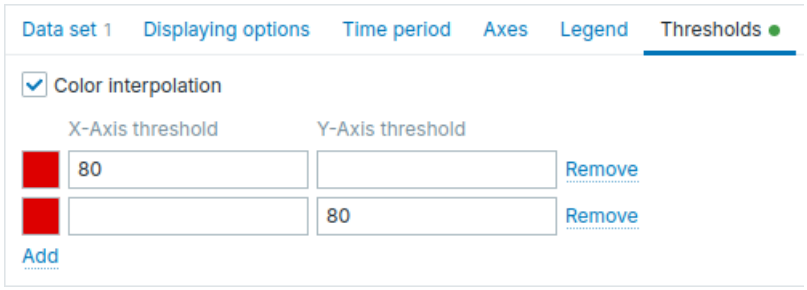
Вкладка **Легенда** позволяет настроить легенду точечной диаграммы:



Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на точечной диаграмме (по умолчанию флажок установлен).
Показывать функцию агрегации	Установите этот флажок, чтобы отображать функцию агрегации в легенде.
Строки	Выберите режим отображения строк легенды: **Фиксированное количество** - количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное количество** - количество отображаемых строк определяется количеством настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.
Количество строк/ Максимальное количество строк	Если для параметра *Строки* установлено значение «Фиксированное количество», задайте количество отображаемых строк легенды (1-10). Если для параметра *Строки* установлено значение «Переменное количество», задайте максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Задайте количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Показывать мин./средн./макс.* не установлен.

Пороговые значения

Вкладка **Пороговые значения** позволяет настроить пороговые значения, определяющие цвета маркеров узлов сети:



Интерполяция цветов	Установите флажок, чтобы включить плавный переход между цветами пороговых значений для маркеров узлов сети. Этот параметр применяется, если для оси X, оси Y или обеих осей одновременно заданы два или более пороговых значения.
Пороговое значение для оси X/оси Y	Нажмите *Добавить*, чтобы добавить пороговое значение, выберите его цвет в палитре и укажите числовое значение. Пороговые значения можно применять к одной оси, к обеим осям по отдельности или к обеим осям одновременно. Например, одно пороговое значение с X=80 и Y=80 окрасит маркеры, если выполнены оба условия. После сохранения список пороговых значений будет отсортирован по возрастанию. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются сопоставления значений.

Отчет SLA

Обзор

Виджет *SLA report* отображает SLA reports, помогая вам отслеживать производительность сервиса в сравнении с заданными целями.

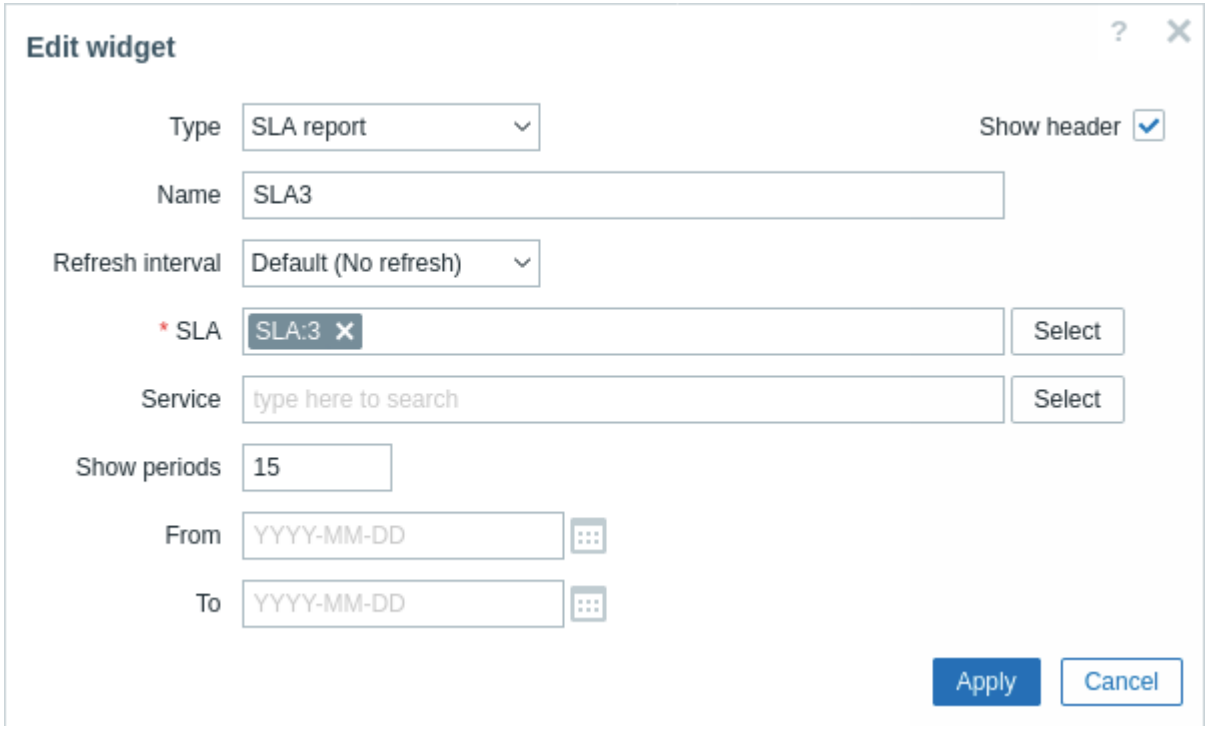
SLA report

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	99.6863	23h 55m 29s	4m 31s	-3m 5s	
2025-08-30	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-08-29	99.9%	99.7095	23h 55m 49s	4m 11s	-2m 45s	
2025-08-28	99.9%	99.9421	23h 59m 10s	50s	36s	
2025-08-27	99.9%	99.7188	23h 55m 57s	3m 15s	-2m 37s	
2025-08-26	99.9%	99.7755	23h 56m 46s	3m 14s	-1m 48s	
2025-08-25	99.9%	99.9094	22h 58m 45s	1h 1m 15s	8s	2025-08-25 01:00 AM Maintenance: 1h

Он отображает те же данные, что и *Services* > *SLA report*.

Конфигурация

Чтобы настроить, выберите *SLA report* в качестве типа:



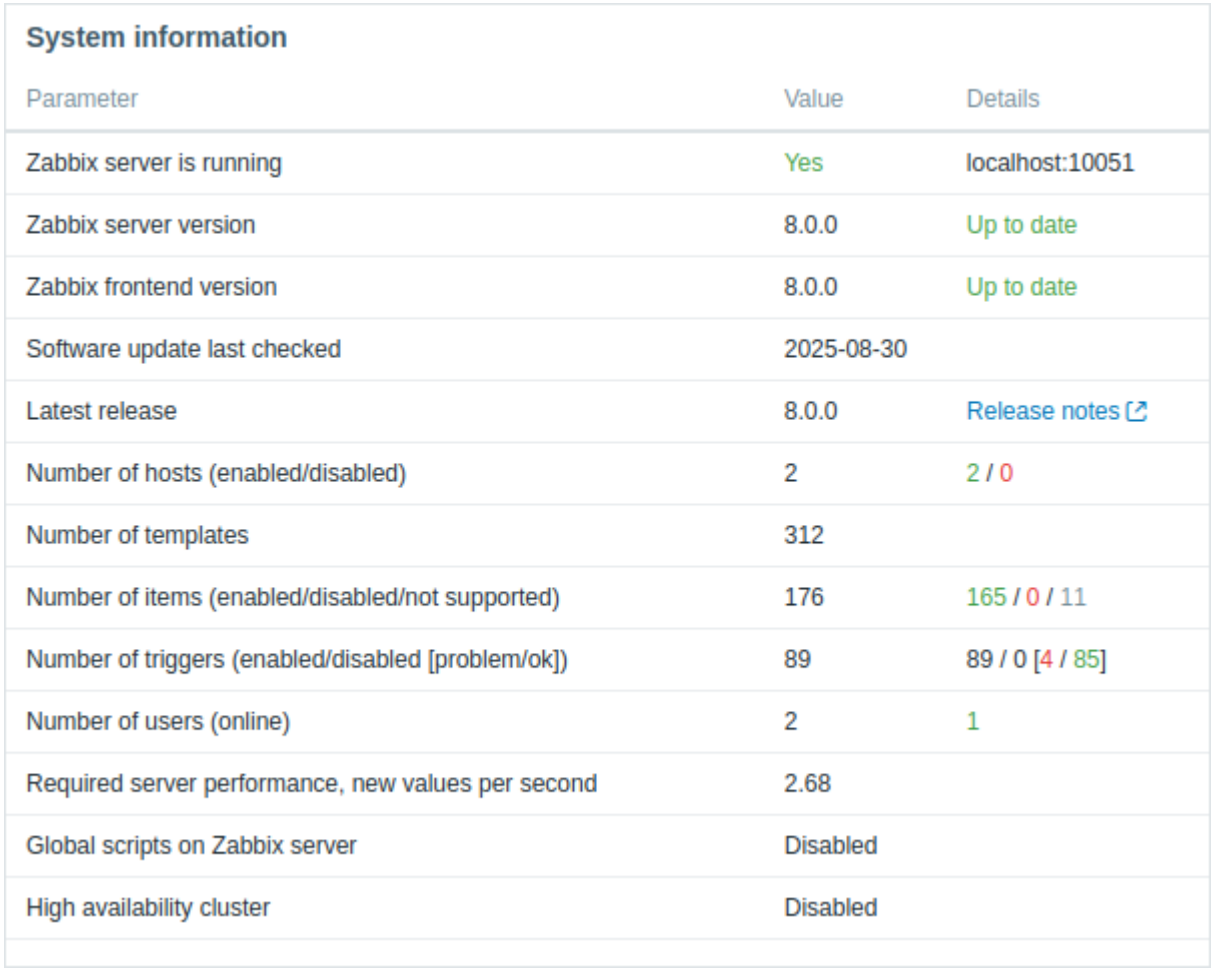
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

SLA	Выберите SLA для отчета.
Service	Выберите сервис для отчета.
Show periods	Укажите, сколько периодов будет отображаться в виджете (по умолчанию 20, максимум 100).
From	Выберите начальную дату для отчета. Поддерживаются относительные даты: `now`, `now/d`, `now/w-1w` и т. д.; поддерживаемые модификаторы даты: d, w, M, y.
To	Выберите конечную дату для отчета. Поддерживаются относительные даты: `now`, `now/d`, `now/w-1w` и т. д.; поддерживаемые модификаторы даты: d, w, M, y.

Системная информация

Обзор

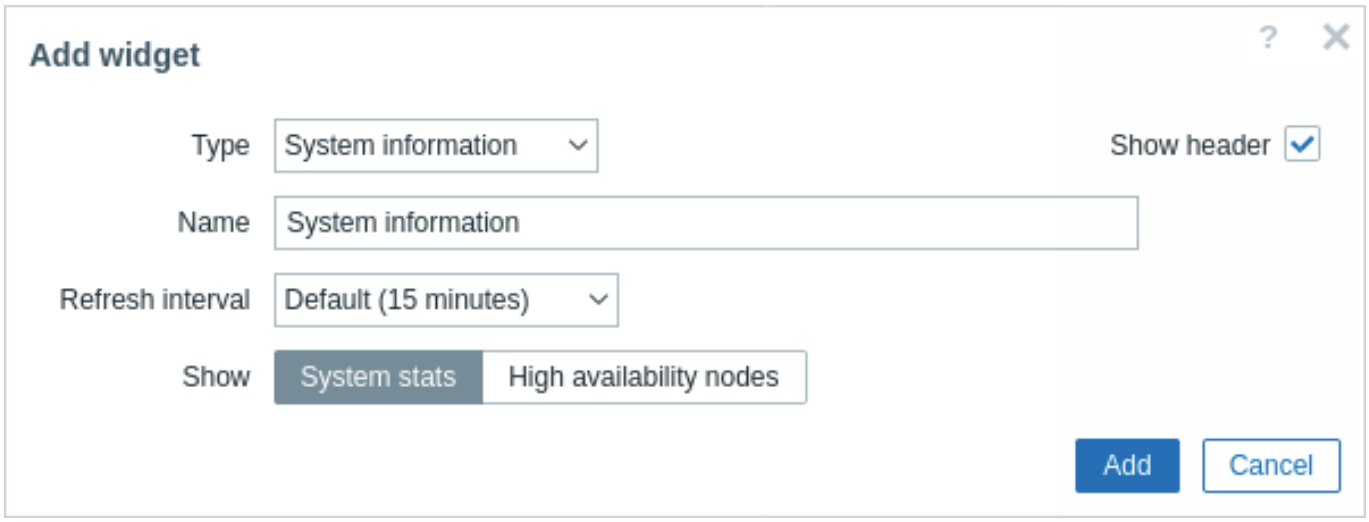
Виджет *System information* отображает сводку основных данных о сервере Zabbix и системе, а при включенной высокой доступности — сведения об узлах высокой доступности.



Он отображает те же данные, что и *Reports* > *System information*, за исключением *Zabbix server ID*.

Настройка

Для настройки выберите тип *Сведения о системе*:



Помимо параметров, общих для всех виджетов, можно задать следующие параметры:

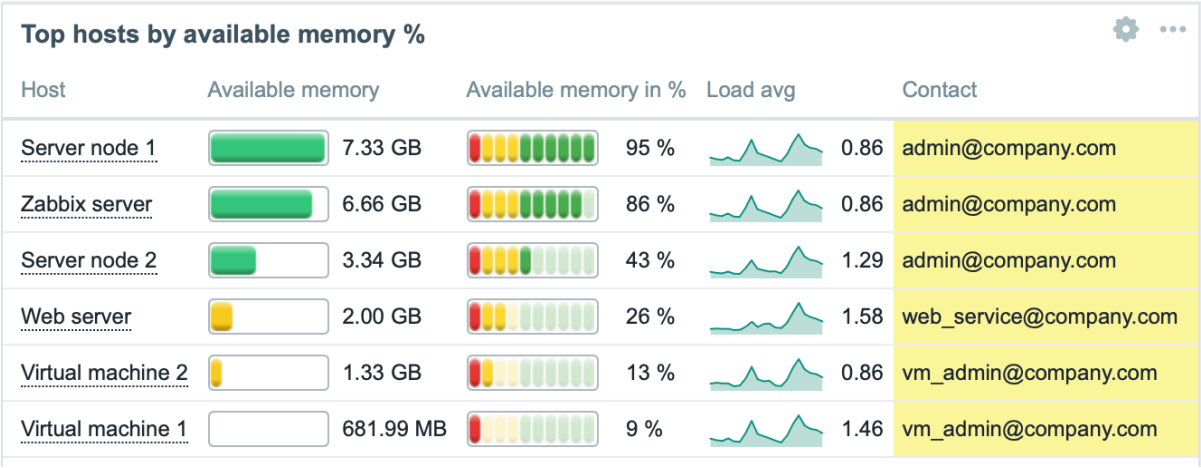
Показывать	Выберите, что отображать: **Статистика системы** - отображать сводку основных данных о сервере Zabbix и системе; **Узлы высокой доступности** - отображать состояние узлов высокой доступности (если включен кластер высокой доступности).

Показывать сведения о проверке обновлений программного обеспечения	Установите флажок, чтобы отображать сведения о проверке обновлений программного обеспечения Zabbix. Этот параметр доступен только в том случае, если проверка обновлений программного обеспечения включена в конфигурации сервера Zabbix и в поле *Показывать* выбрано значение «Статистика системы».

Топ узлов сети

Обзор

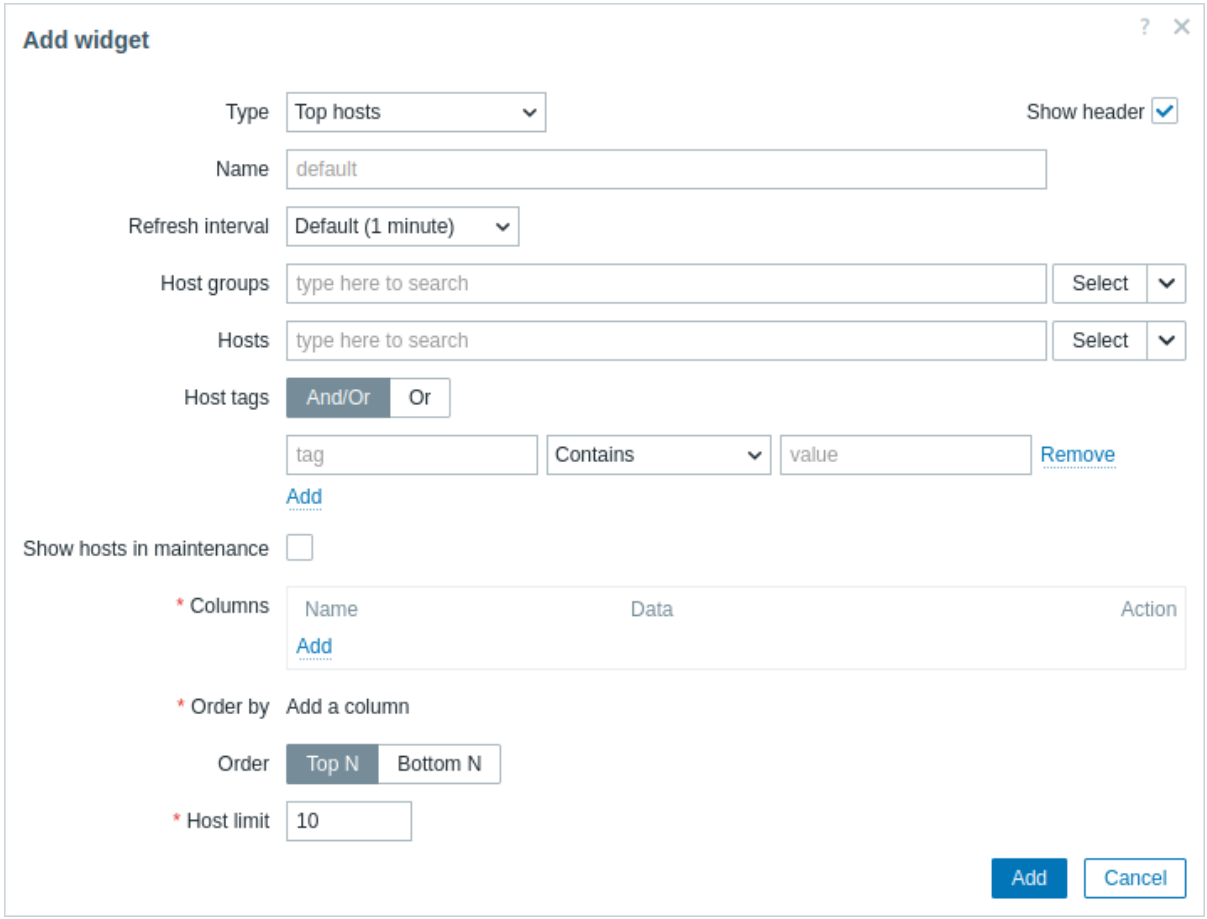
Виджет *Top hosts* отображает наибольшие/наименьшие значения для выбранных элементов данных с нескольких узлов сети.



Виджет поддерживает отображение до 1000 узлов сети одновременно.

Настройка

Для настройки выберите *Топ узлов сети (Top hosts)* в качестве типа:



Помимо параметров, которые являются общими для всех виджетов, можно задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети для отображения в виджете. В качестве альтернативы выберите совместимый виджет как источник данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на шаблонной панели.
Узлы сети	Выберите узлы сети для отображения в виджете. В качестве альтернативы выберите совместимый виджет или панель как источник данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Теги узлов сети	Укажите теги, чтобы ограничить количество узлов сети, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно выполнения одного условия. Этот параметр недоступен при настройке виджета на шаблонной панели.

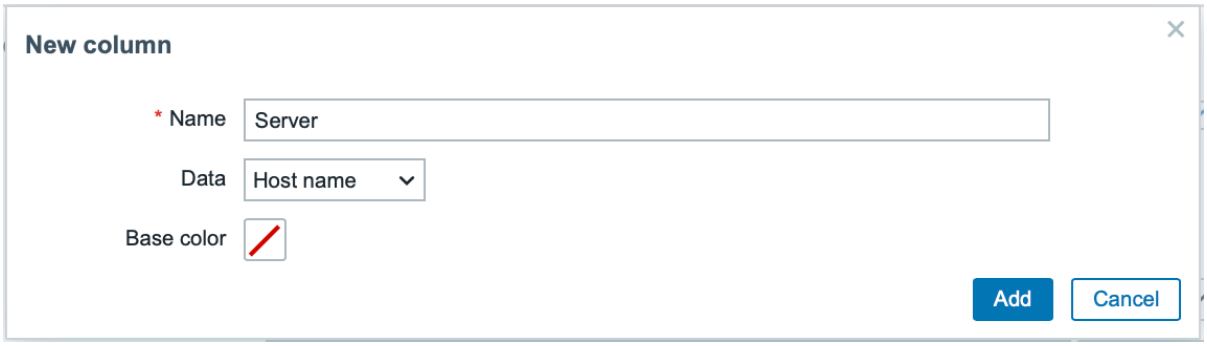
Показывать узлы сети в обслуживании	Отметьте этот флажок, чтобы также отображались узлы сети в обслуживании (в этом случае рядом с именем узла сети будет показан значок обслуживания). По умолчанию не отмечен.
Столбцы	Добавьте столбцы данных для отображения. Порядок столбцов определяет их отображение слева направо. Столбцы можно менять местами, перетаскивая их вверх и вниз за маркер перед именем столбца.
Сортировать по	Укажите столбец из заданного списка *Columns*, который будет использоваться для сортировки *Top N* или *Bottom N*.
Порядок	Укажите порядок строк: **Top N** - по убыванию в соответствии с агрегированным значением *Order by*; **Bottom N** - по возрастанию в соответствии с агрегированным значением *Order by*.
Ограничение по узлам сети	Количество строк узлов сети для отображения (1-1000). Этот параметр недоступен при настройке виджета на шаблонной панели.

Настройка столбцов

Можно добавить столбцы трех типов данных: имя узла сети, значение элемента данных или текст. Список доступных параметров столбца зависит от типа данных столбца, а для типа значения элемента данных — от формата значения.

Столбец Host name

Столбец Host name используется для отображения имени узла сети.



Поддерживаемые параметры:

Name	Имя столбца.
Data	Тип данных для отображения в столбце, выберите **Host name**.
Base color	Цвет фона столбца.

Текстовый столбец

Текстовый столбец используется для отображения любой указанной текстовой строки.



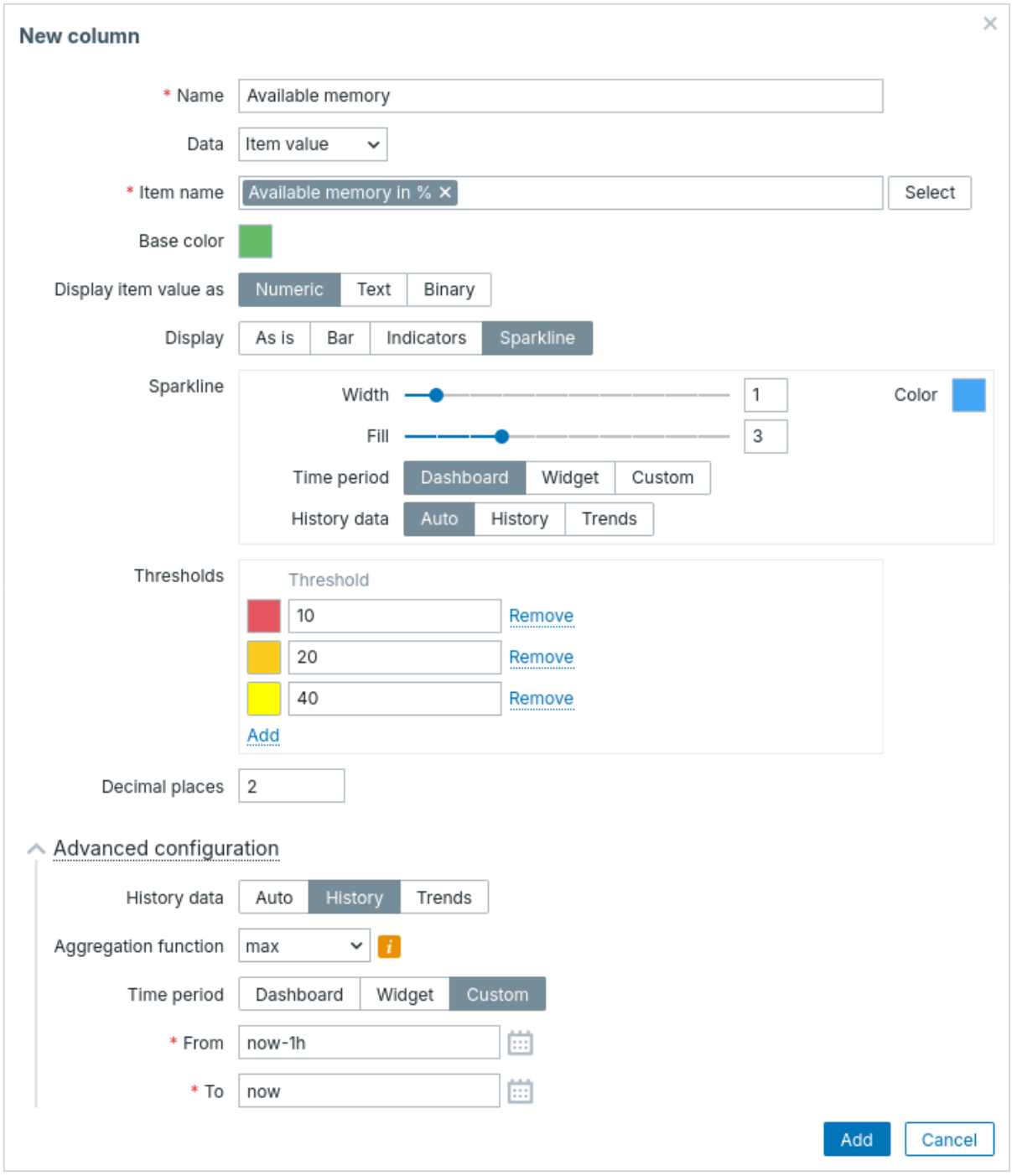
Поддерживаемые параметры:

Name	Название столбца.
Data	Тип данных для отображения в столбце; выберите **Text**.
Base color	Цвет фона столбца.

Text	Введите строку для отображения. Может содержать макросы узла сети и инвентарных данных.

Столбец значения элемента данных

Столбец значения элемента данных используется для отображения значения указанного элемента данных; поддерживается несколько форматов и вариантов отображения значений.



Поддерживаемые параметры:

Name	Имя столбца.
Data	Тип данных для отображения в столбце, выберите **Item value**.

Item name	Выберите элемент данных; его имя используется для поиска и отображения элементов данных с одинаковым именем на всех выбранных узлах сети. Если на узле сети есть несколько таких элементов данных, виджет отображает элемент данных, ключ которого первым идет в алфавитном порядке (например, *CPU utilization* с ключом `proc.cpu.util`, а не `system.cpu.util`). При настройке виджета на панели шаблона можно выбрать только элементы данных, настроенные в шаблоне.
Base color	Цвет фона столбца; цвет заливки, если данные *Item value* отображаются в виде полосы или индикаторов. Цвет по умолчанию можно заменить пользовательским цветом, если значение элемента данных превышает один из указанных порогов.
Display item value as	Формат отображения значения элемента данных: **Numeric**, **Text** или **Binary**. Выбранный вариант определяет доступные дополнительные параметры. Список параметров для каждого формата приведен ниже.
Advanced configuration	
History data	Источник данных: история или тренды: **Auto** - автоматический выбор; **History** - данные истории; **Trends** - данные трендов. Этот параметр доступен только для числовых значений элементов данных. Для текстовых и двоичных значений данные всегда берутся из истории.
Aggregation function	Укажите функцию агрегации, которую следует использовать в пределах выбранного *Time period*: **min** - отображать наименьшее значение; **max** - отображать наибольшее значение; **avg** - отображать среднее значение; **count** - отображать количество значений; **sum** - отображать сумму значений; **first** - отображать первое значение; **last** - отображать последнее значение. Если выбрано **not used**, отображается последнее значение без агрегации. Агрегация позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день) вместо последнего значения. Для *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для *count* нечисловые данные преобразуются в числовые.
Time period	Укажите период времени, используемый для агрегации значений. Выберите источник данных для периода времени: **Dashboard** - использовать селектор периода времени панели; **Widget** - использовать совместимый виджет (указанный в параметре *Widget*); **Custom** - использовать пользовательский период времени, заданный в параметрах *From* и *To*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении указателя мыши. Обратите внимание: независимо от настройки *Time period* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени. Этот параметр недоступен, если для *Aggregation function* выбрано значение ”not used”.
Widget	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Time period* выбрано значение ”Widget”.
From	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* выбрано значение ”Custom”.
To	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* выбрано значение ”Custom”.

Параметры числового значения элемента данных:

Display	Определите способ отображения значения: **As is** - обычный текст; **Bar** - сплошная полоса с цветной заливкой; **Indicators** - сегментированная полоса с цветной заливкой; **Sparkline** - миниатюрный линейный график.
Thresholds	Укажите пороговые значения, при достижении которых должен изменяться цвет фона или заливки. При сохранении список будет отсортирован по возрастанию. Для спарклайнов пороги применяются только к последнему значению элемента данных.
Decimal places	Укажите количество десятичных знаков, отображаемых со значением.
Bar/Indicators	
Min	Минимальное значение.
Max	Максимальное значение.
Sparkline	
Width	Задайте толщину линии графика с помощью ползунка или вручную введите значение от 0 до 10.
Color	Выберите цвет линии и заливки.
Fill	Задайте уровень прозрачности заливки с помощью ползунка или вручную введите значение от 0 до 10.
Time period	Укажите период времени, значения за который будут включены в график спарклайна. Выберите источник данных для периода времени: **Dashboard** - использовать селектор периода времени панели; **Widget** - использовать совместимый виджет (указанный в параметре *Widget*); **Custom** - использовать пользовательский период времени, заданный в параметрах *From* и *To*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении указателя мыши. Обратите внимание: независимо от настройки *Time period* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени.
Widget	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Time period* выбрано значение "Widget".
From	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* выбрано значение "Custom".
To	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Time period* выбрано значение "Custom".
History data	Источник данных: история или тренды: **Auto** - автоматический выбор; **History** - данные истории; **Trends** - данные трендов.

Параметры текстового значения элемента данных:

Highlights	Укажите регулярные выражения, при совпадении с которыми должен изменяться цвет фона или заливки.

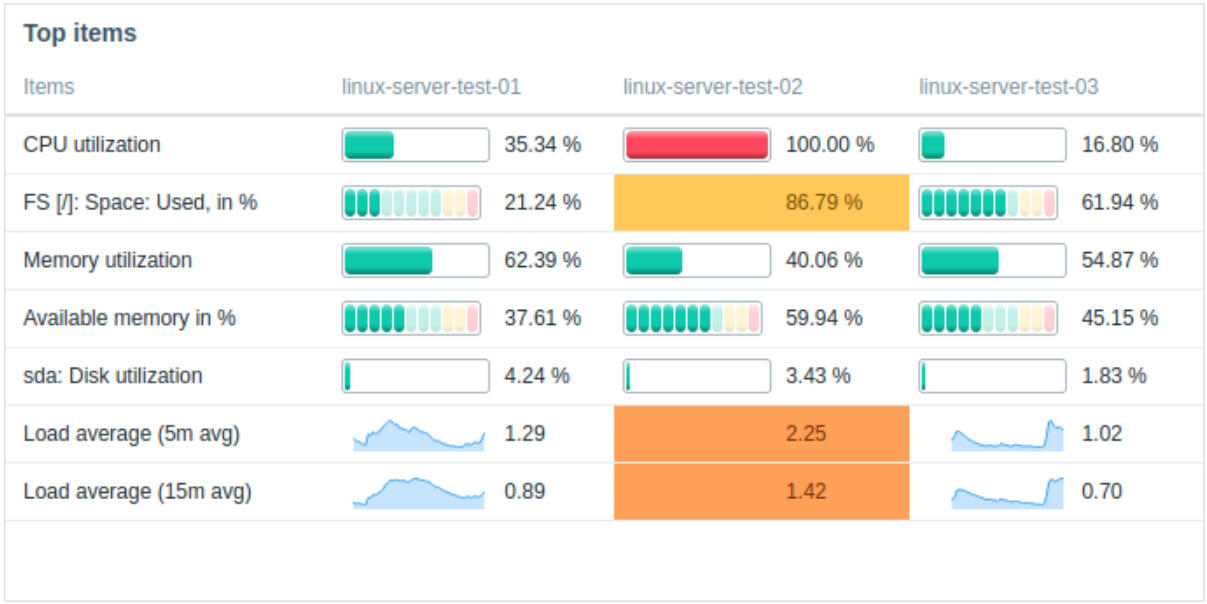
Параметры двоичного значения элемента данных:

Show thumbnail	Укажите, следует ли создавать и отображать миниатюру изображения, содержащего двоичные данные, или отображать в столбце значений гиперссылку *Show*, ведущую к полноразмерному изображению.

Топ элементов данных

Обзор

Виджет *Top items* отображает наибольшие/наименьшие значения для выбранных элементов данных, обеспечивая быстрый обзор их производительности.

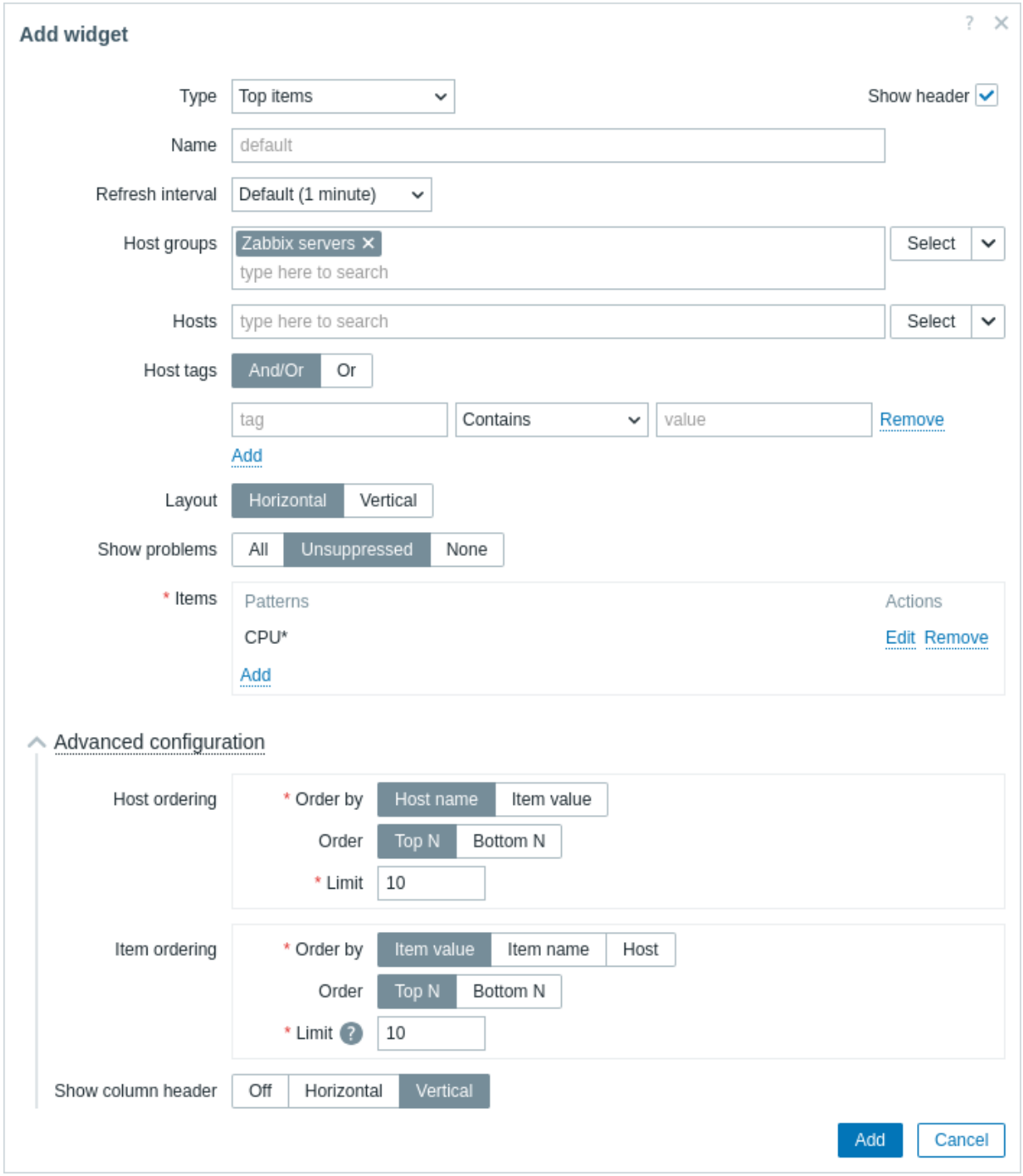


По умолчанию виджет отображает значения за последние 24 часа. Этот период можно изменить в разделе *Администрирование* > *Общие* > *GUI* (*Максимальный период отображения истории*).

Щелчок по значению элемента данных открывает меню элемента данных.

Конфигурация

Чтобы настроить, выберите *Top items* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специфические параметры:

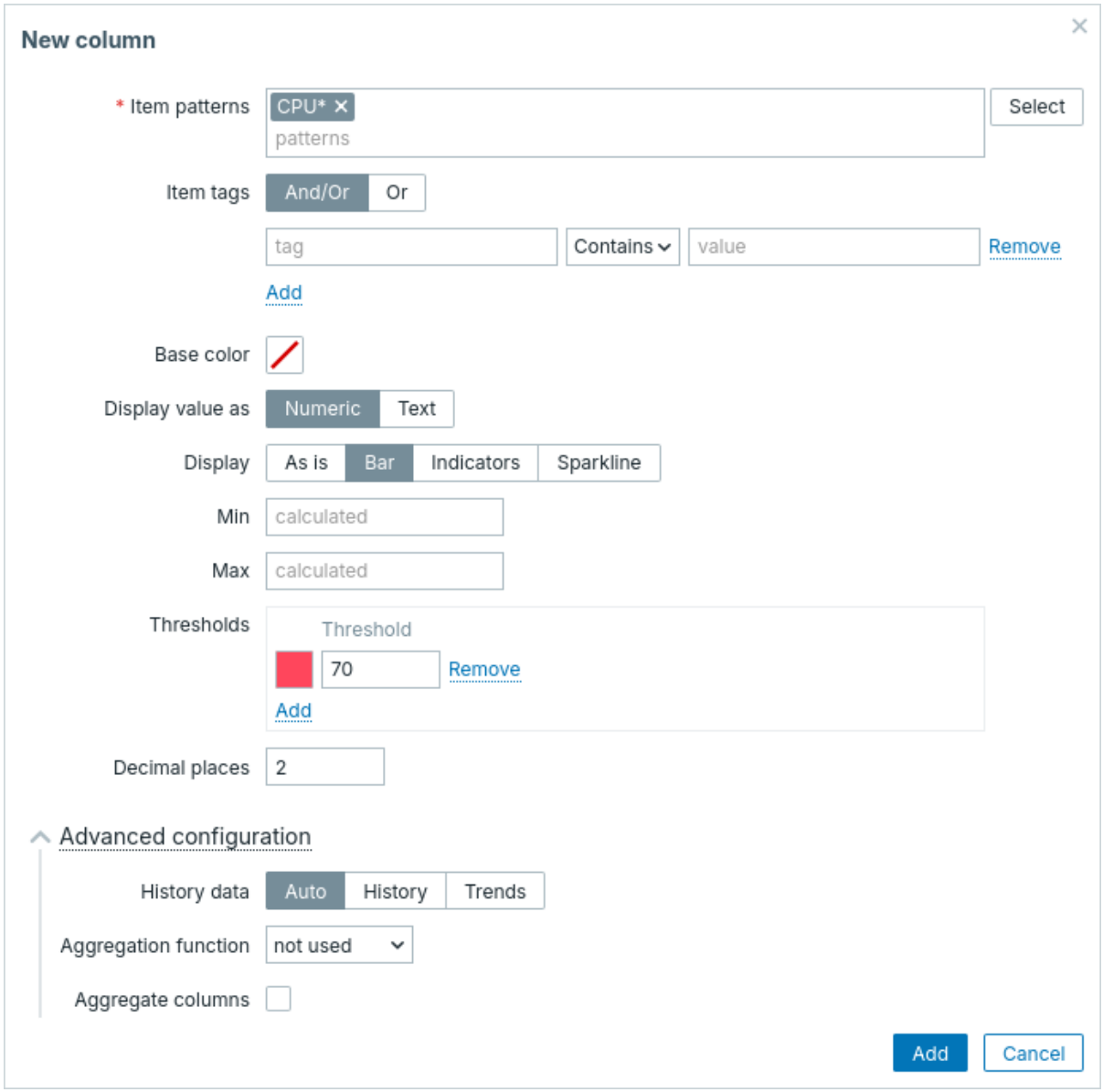
Группы узлов сети	Выберите группы узлов сети. Кроме того, можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на шаблонной панели.
Узлы сети	Выберите узлы сети. Кроме того, можно выбрать совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.

Теги узлов сети	Укажите теги, чтобы ограничить количество данных узлов сети, отображаемых в виджете. Можно включать и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия.
Макет	Выберите вариант макета: **Horizontal** - имена узлов сети будут отображаться слева; **Vertical** - имена узлов сети будут отображаться сверху.
Показывать проблемы	Отфильтруйте отображаемые проблемы по их статусу (all, unsuppressed, none). Проблемы обозначаются цветными значениями элементов данных, переопределяя настройки отображения столбца элемента данных. Цвета основаны на серьезности проблемы, которую можно настроить в *Administration* > *General* > *Trigger displaying options*. Серьезность проблемы можно изменить при обновлении проблем.
Элементы данных	Добавьте шаблоны элементов данных или конкретные элементы данных для отображения (см. Настройка столбцов).
Дополнительная конфигурация	
Порядок узлов сети	Выберите параметры сортировки для столбца/строки узлов сети.
Сортировать по	Сортировать узлы сети по: **Host name** - узлы сети будут отсортированы по имени узла сети; **Item value** - узлы сети будут отсортированы по значению выбранных элементов данных.
Порядок	Выберите, отображать ли наибольшие или наименьшие значения: **TopN** - первые N значений; **BottomN** - последние N значений. Значение N выбирается в поле *Limit*.
Ограничение	Введите ограничение для отображаемых узлов сети (диапазон 1-1000; по умолчанию 10). Это значение станет значением N в поле *Order*.
Порядок элементов данных	Выберите параметры сортировки для столбца/строки элементов данных.
Сортировать по	Сортировать элементы данных по: **Item value** - элементы данных будут отсортированы по значению элемента данных; **Item name** - элементы данных будут отсортированы по имени элемента данных; **Host** - элементы данных будут отсортированы по выбранному шаблону узла сети.
Порядок	Выберите, отображать ли наибольшие или наименьшие значения: **TopN** - первые N значений; **BottomN** - последние N значений. Значение N выбирается в поле *Limit*.
Ограничение	Введите ограничение для отображаемых элементов данных (диапазон 1-1000; по умолчанию 10). Это значение станет значением N в поле *Order*.

Показывать заголовок столбца	Выберите параметры отображения заголовка столбца: **Off** - не отображать заголовок столбца; **Horizontal** - отображать текст в заголовке горизонтально; **Vertical** - отображать текст в заголовке вертикально.

Настройка столбцов

Чтобы настроить столбцы элементов данных, нажмите *Добавить* в параметре *Элементы данных*:



Общие параметры столбцов:

Шаблоны элементов данных	Укажите один или несколько шаблонов элементов данных, используя подстановочный знак. Либо выберите элементы данных. При настройке виджета на дашборде шаблона можно выбирать только элементы данных, настроенные в шаблоне.

Теги элементов данных	Укажите теги, чтобы ограничить количество данных элементов данных, отображаемых в виджете. Можно как включать, так и исключать определённые теги и значения тегов. Можно задать несколько условий. Сопоставление имён тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена тегов и значения (с учётом регистра); **Содержит** — включить указанные имена тегов, где значения тегов содержат введённую строку (поиск подстроки, без учёта регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена тегов и значения (с учётом регистра); **Не содержит** — исключить указанные имена тегов, где значения тегов содержат введённую строку (поиск подстроки, без учёта регистра). Для условий доступны два типа вычисления: **И/Или** — должны быть выполнены все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию *Или*; **Или** — достаточно выполнения одного условия.
Базовый цвет	Выберите цвет фона столбца или цвет заливки, если для параметра *Отображение* установлено значение ”Полоса” или ”Индикаторы”. Обратите внимание, что базовый цвет может быть переопределён цветами порогов или подсветки.
Отображать значение как	Формат отображения значения элемента данных: **Числовой** или **Текст**. Выбранный вариант определяет, какие дополнительные параметры будут доступны. Ниже приведён список параметров для каждого формата.

Параметры столбца для числовых значений:

Отображение	Определите, как должно отображаться значение: **Как есть** — обычный текст; **Полоса** — сплошная полоса, залитая цветом; **Индикаторы** — сегментированная полоса, залитая цветом; **Спарклайн** — мини-график в виде линии.
Мин.	Минимальное значение для отображения полосы/индикаторов.
Макс.	Максимальное значение для отображения полосы/индикаторов.
Ширина	Задайте толщину линии графика с помощью ползунка или вручную введите значение в диапазоне от 0 до 10. Этот параметр доступен только для отображения спарклайна.
Заливка	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введите значение в диапазоне от 0 до 10. Этот параметр доступен только для отображения спарклайна.
Цвет	Выберите цвет линии и заливки. Этот параметр доступен только для отображения спарклайна.
Описание полей, связанных с выбором периода времени и исторических данных, см. в разделе *Расширенная настройка*.	
Пороговые значения	Укажите пороговые значения, при достижении которых должен изменяться цвет фона/заливки. При сохранении список будет отсортирован по возрастанию. Для спарклайнов пороговые значения применяются только к последнему значению элемента данных.
Десятичные знаки ***Расширенная настройка***	Укажите количество десятичных знаков, отображаемых со значением.

Исторические данные	Получать данные из истории или трендов: **Автоматически** — автоматический выбор; **История** — получать исторические данные; **Тренды** — получать данные трендов. Этот параметр доступен только для числовых значений элементов данных. Для текстовых значений данные всегда будут получаться из истории.
Функция агрегации	Укажите функцию агрегации, которую следует использовать в выбранном *Периоде времени*: **min** — отображать наименьшее значение; **max** — отображать наибольшее значение; **avg** — отображать среднее значение; **count** — отображать количество значений; **sum** — отображать сумму значений; **first** — отображать первое значение; **last** — отображать последнее значение. Если выбрано **не используется**, отображается самое последнее значение (без агрегации). Агрегация позволяет отображать агрегированное значение за указанный интервал (5 минут, час, день) вместо самого последнего значения. Для функций *min*, *max*, *avg* и *sum* можно отображать только числовые данные. Для функции *count* нечисловые данные будут преобразованы в числовые.
Период времени	Укажите период времени, используемый для агрегации значений. Выберите источник данных для периода времени: **Панель мониторинга** — использовать селектор периода времени панели мониторинга; **Виджет** — использовать совместимый виджет (указанный в параметре *Виджет*); **Пользовательский** — использовать пользовательский период времени, заданный в параметрах *От* и *До*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении указателя мыши. Обратите внимание, что независимо от настройки *Периода времени* виджета совместимые виджеты по-прежнему могут использовать его как источник данных для периода времени. Этот параметр недоступен, если для *Функции агрегации* установлено значение «не используется».
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Периода времени* установлено значение «Виджет».
От	Введите или выберите начало периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Периода времени* установлено значение «Пользовательский».
До	Введите или выберите конец периода времени. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w` и т. д.). Этот параметр доступен, если для *Периода времени* установлено значение «Пользовательский».
Агрегировать столбцы	Если флажок установлен, все элементы данных, соответствующие шаблонам элементов данных столбца, будут сгруппированы в один объединенный столбец/строку. Значения соответствующих элементов данных будут агрегироваться с использованием выбранной *Функции агрегации столбца* и отображаться под *Именем объединенного столбца*. По умолчанию: выключено (флажок не установлен).Этот параметр отображается только если для *Отображать значение как* выбрано значение **Числовое**.
Функция агрегации столбца	Выберите способ объединения значений из сгруппированных шаблонов элементов данных. Доступные варианты: **min**, **max**, **avg**, **count**, **sum** (по умолчанию). Этот параметр отображается только если установлен флажок *Агрегировать столбцы*.
Имя объединенного столбца	Введите отображаемое имя агрегированного столбца. Это поле является обязательным, если установлен флажок *Агрегировать столбцы*, и становится заголовком столбца для объединенных значений.

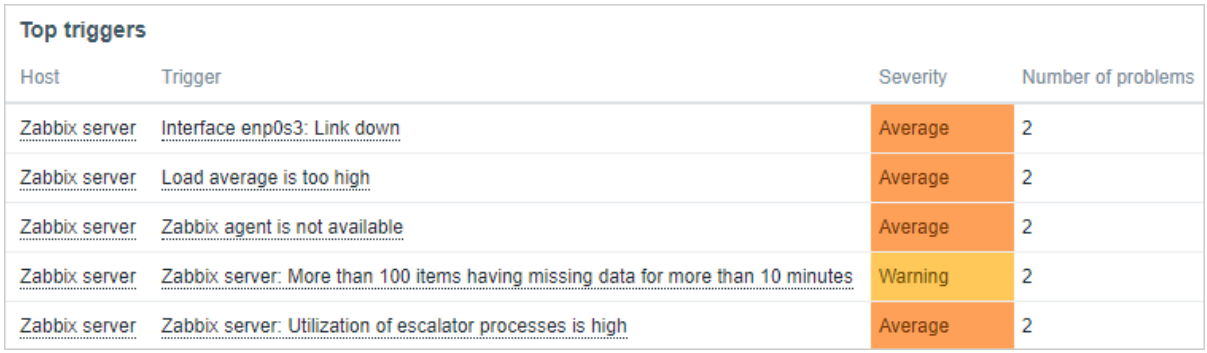
Параметры столбца для текстовых значений:

Выделение	Укажите регулярные выражения, при совпадении с которыми должен изменяться цвет фона/заливки.

Топ триггеров

Обзор

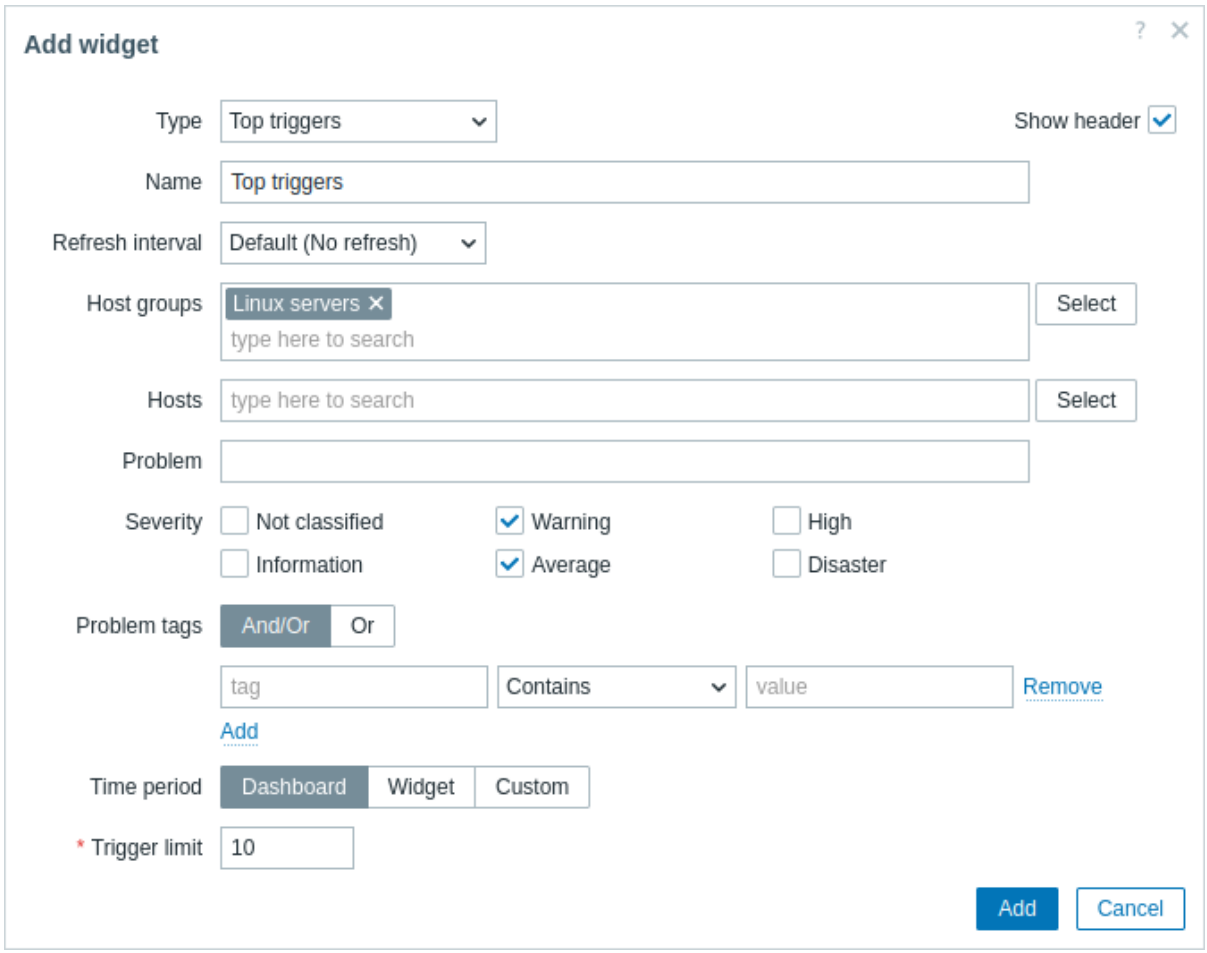
Виджет *Top triggers* отображает триггеры с наибольшим количеством проблем.



Он отображает те же данные, что и *Reports > Top 100 triggers*, и может отображать до 1000 записей.

Конфигурация

Для настройки выберите *Top triggers* в качестве типа виджета:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Host groups	Выберите группы узлов сети, проблемы которых нужно отображать в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Проблемы из этих групп узлов сети будут отображаться в виджете; если группы узлов сети не указаны, будут отображаться проблемы из всех групп узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Hosts	Выберите узлы сети, проблемы которых нужно отображать в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться проблемы всех узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Problem	Можно просматривать только триггеры для определенных проблем. Для этого введите строку, которая должна совпадать с именем проблемы. Макросы не раскрываются.
Severity	Отметьте уровни серьезности триггеров, чтобы отфильтровать триггеры, отображаемые в виджете. Если ни один уровень серьезности не отмечен, будут отображаться все триггеры.
Problem tags	Укажите теги проблем, которые нужно отображать в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия.
Time period	Выберите источник данных для временного периода: **Dashboard** - использовать селектор временного периода панели; **Widget** - использовать совместимый виджет (указанный в параметре *Widget*); **Custom** - использовать пользовательский временной период, заданный в параметрах *From* и *To*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий заданное время при наведении курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты по-прежнему могут использовать его как источник данных для временного периода.
Widget	Введите или выберите совместимый виджет в качестве источника данных для временного периода. Этот параметр доступен, если для *Time period* установлено значение ”Widget”.
From	Введите или выберите начало временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w`, и т. д.). Этот параметр доступен, если для *Time period* установлено значение ”Custom”.
To	Введите или выберите конец временного периода. Поддерживается синтаксис относительного времени (`now`, `now/d`, `now/w-1w`, и т. д.). Этот параметр доступен, если для *Time period* установлено значение ”Custom”.
Trigger limit	Установите количество триггеров, которые будут отображаться. Диапазон допустимых значений: 1-1000.

Обзор триггеров

Обзор

Виджет *Обзор триггеров* отображает текущие состояния триггеров узлов сети в виде таблицы цветных блоков. Он предоставляет быстрый визуальный снимок состояния и активности триггеров на разных узлах сети.



Виджет использует несколько визуальных элементов для отображения сведений о триггерах:

- Цвета блоков для триггеров PROBLEM соответствуют уровням серьезности проблем.
- Мигающие блоки указывают на недавние изменения состояния триггера (в течение последних 2 минут).
- Серые стрелки указывают на триггеры с зависимостями (наведите на них курсор, чтобы увидеть подробности).
- Значки флажков указывают на подтвержденные проблемы; чтобы этот значок появился, все активные или решенные проблемы для триггера должны быть подтверждены.

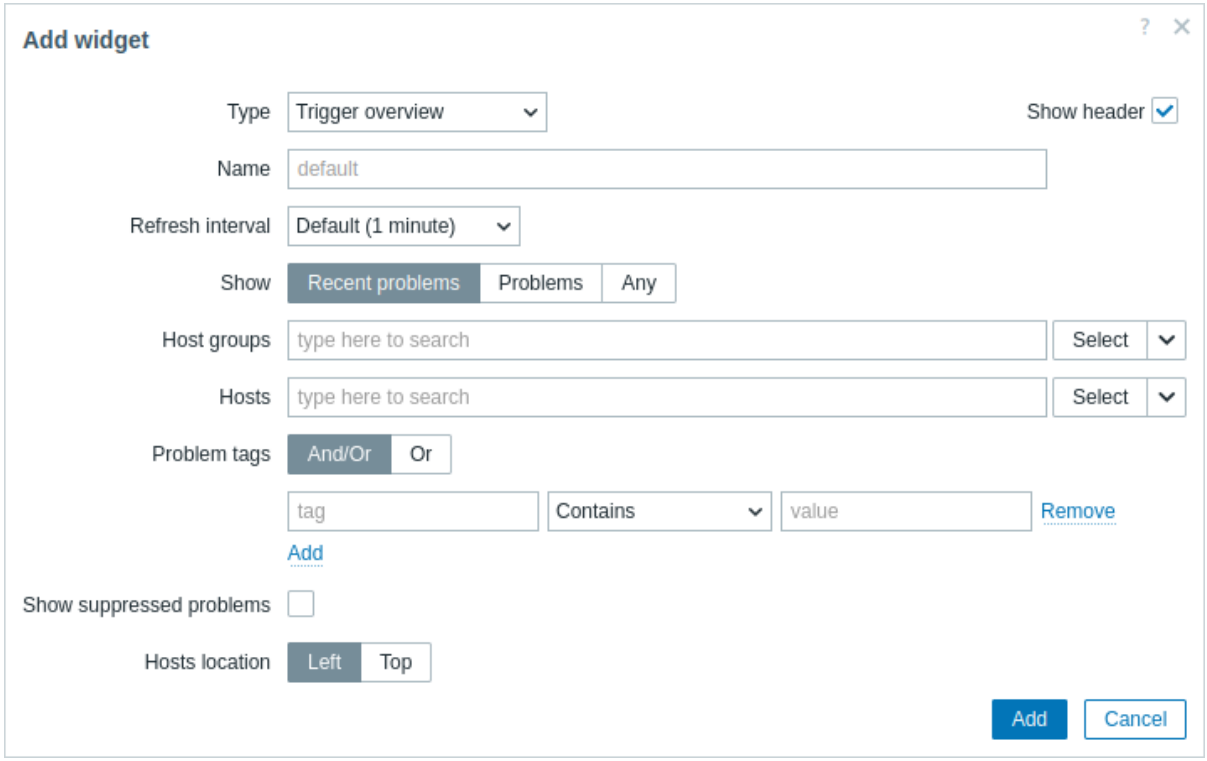
Note:
Цвета серьезности триггеров можно настроить в *Администрирование* > *Общее*. Серьезность триггеров можно изменять при обновлении проблем.

При нажатии на блок отображаются ссылки на сведения, связанные с триггером (проблемы, конфигурация триггера и т. д.).

По умолчанию отображается до 50 записей (настраивается в *Администрирование* > *Общее* > *GUI* в разделе *Максимальное число столбцов и строк в таблицах обзора*). Если триггеров больше, выводится сообщение с предложением уточнить критерии фильтрации. Это ограничение применяется до любой дополнительной фильтрации (например, по тегам).

Конфигурация

Чтобы настроить, выберите *Trigger overview* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

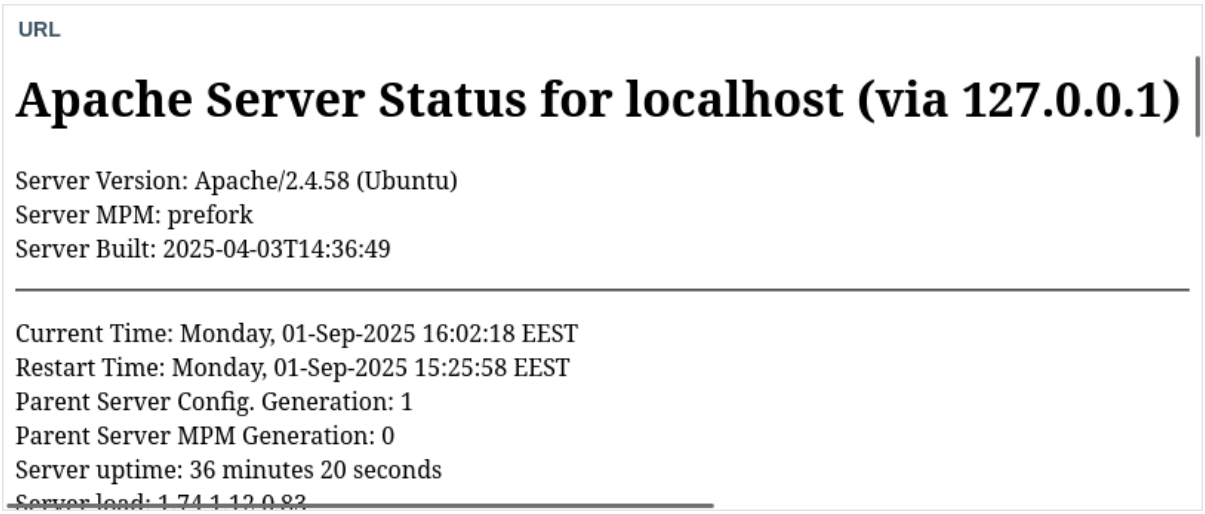
Show	Фильтровать триггеры по состоянию триггера: **Recent problems** - *(по умолчанию)* показывать триггеры, которые недавно были или все еще находятся в состоянии PROBLEM (устраненные и неустраненные); **Problems** - показывать триггеры, которые находятся в состоянии PROBLEM (неустраненные); **Any** - показывать все триггеры.
Host groups	Выберите группы узлов сети. Либо выберите совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предлагаться выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на панели шаблона.
Hosts	Выберите узлы сети. Либо выберите совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предлагаться выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Problem tags	Укажите теги для фильтрации триггеров, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. **Примечание:** Если параметр *Show* установлен в 'Any', будут отображаться все триггеры, даже если указаны теги. Однако, хотя последние изменения состояния триггера (отображаемые как мигающие блоки) будут обновляться для всех триггеров, сведения о состоянии триггера (цвет серьезности проблемы и наличие подтверждения проблемы) будут обновляться только для триггеров, соответствующих указанным тегам. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия.
Show suppressed problems	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
Layout	Выберите вариант размещения: **Horizontal** - имена узлов сети будут отображаться слева; **Vertical** - имена узлов сети будут отображаться сверху.

URL

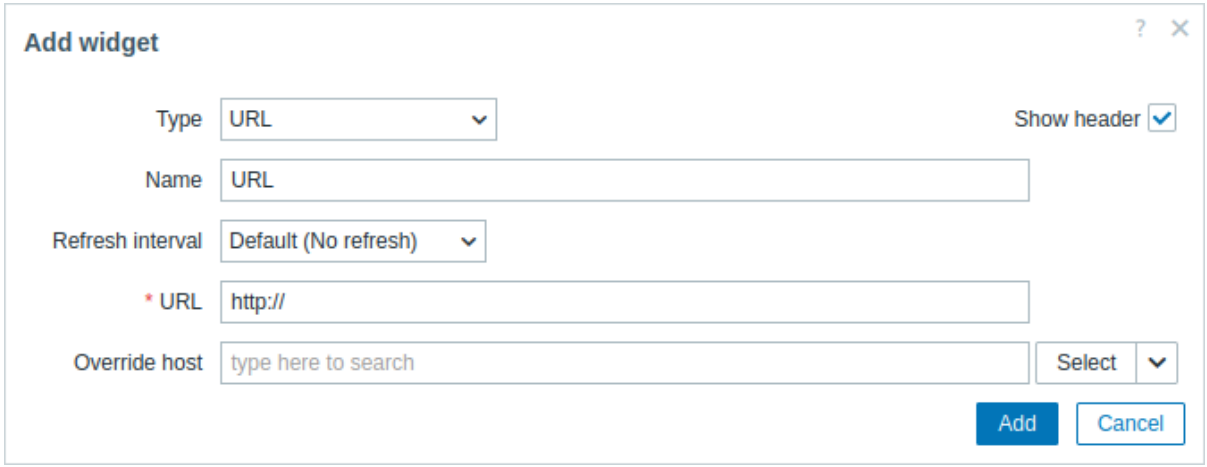
Обзор

Виджет *URL* отображает содержимое, полученное по URL.



Конфигурация

Чтобы настроить, выберите *URL* в качестве типа:



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

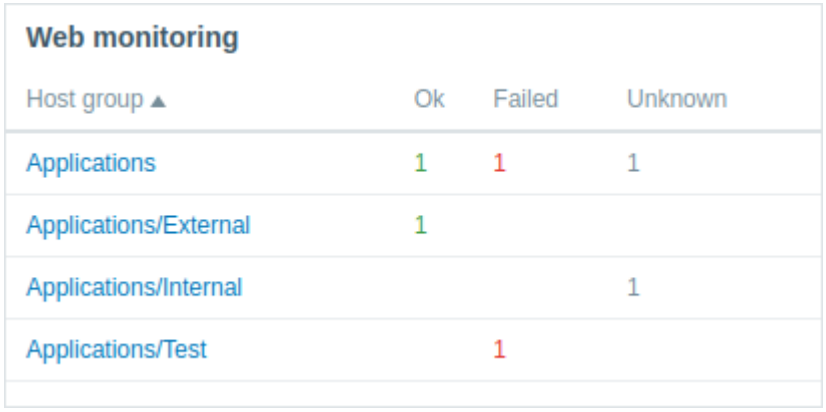
URL	Введите URL для отображения (до 2048 символов). Внешние URL должны начинаться с `http://` или `https://`. Внутренние URL поддерживают относительные пути (например, `zabbix.php?action=report.status`). Поддерживаются макросы `{HOST.*}`.
Override host	Выберите совместимый виджет или селектор узлов сети на веб-интерфейсе панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.

Attention:
Браузеры могут не загружать HTTP-страницу, настроенную в виджете, если веб-интерфейс Zabbix открыт по HTTPS. Также может потребоваться изменить HTTP-заголовок ответа X-Frame-Options, чтобы отображать содержимое, внешнее по отношению к веб-интерфейсу Zabbix.

Веб-мониторинг

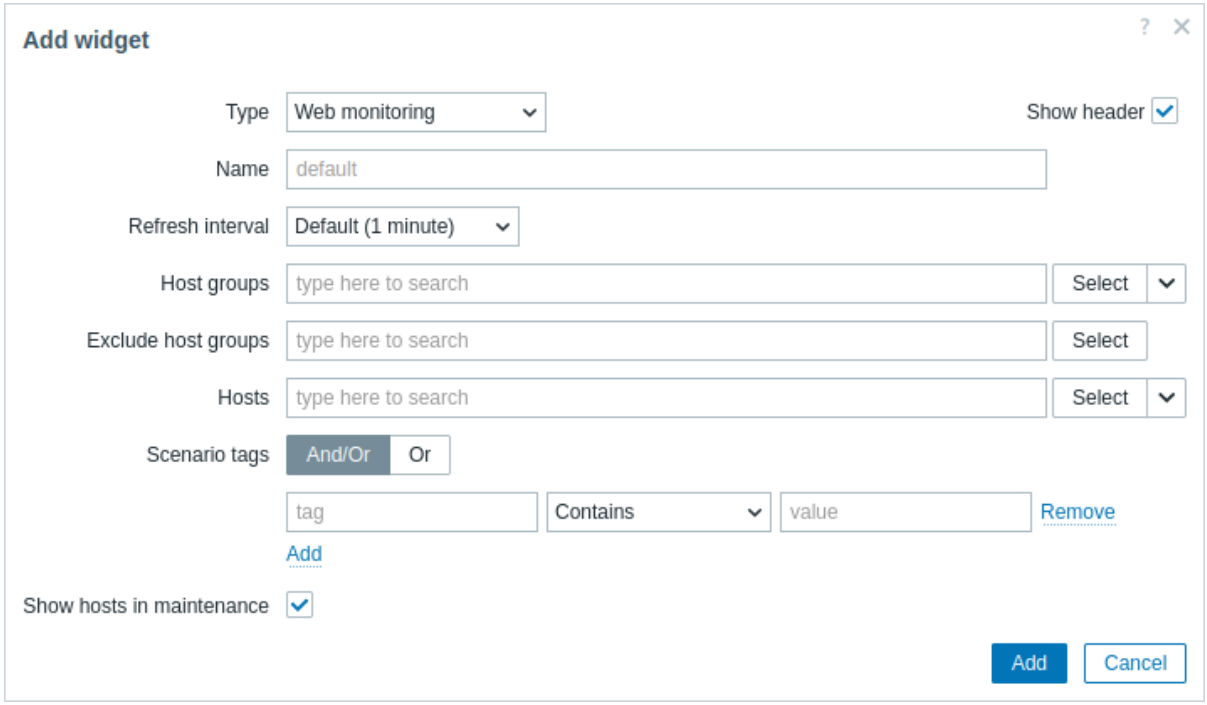
Обзор

Виджет *Web monitoring* отображает статус активных веб-сценариев.



При нажатии на имя группы узлов сети открывается *Monitoring* > *Hosts* > *Web scenarios*, где можно просмотреть сведения о сценарии.

Конфигурация



Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети, которые нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет отображаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп будут отображаться в виджете; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на шаблонной панели.
Исключить группы узлов сети	Выберите группы узлов сети, которые нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет отображаться выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп не будут отображаться. Например, узлы сети 001, 002, 003 могут входить в группу A, а узлы сети 002, 003 также в группу B. Если одновременно выбрать *показать* группу A и *исключить* группу B, на панели будут отображаться только данные узла сети 001. Этот параметр недоступен при настройке виджета на шаблонной панели.
Узлы сети	Выберите узлы сети, которые нужно отображать в виджете. Кроме того, можно выбрать совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет отображаться выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на шаблонной панели.

Теги сценариев	Укажите теги, чтобы ограничить число веб-сценариев, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию *Or*; **Or** - достаточно выполнения одного условия.
Показывать узлы сети в обслуживании	Включать в статистику узлы сети, находящиеся в обслуживании. При настройке виджета на шаблонной панели этот параметр называется *Показывать данные в обслуживании*.

Мониторинг

Обзор

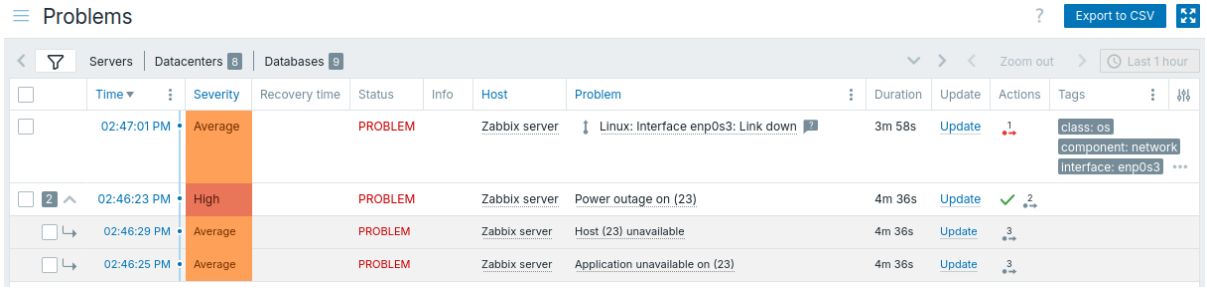
Меню «Мониторинг» предназначено для отображения данных.
Вся информация, которую настроено собирать, визуализировать и обрабатывать в Zabbix, отображается в различных разделах меню *Мониторинг*.

Проблемы

Обзор

В *Monitoring* > *Problems* можно увидеть, какие проблемы у вас сейчас есть. Проблемы — это те триггеры, которые находятся в состоянии ”Problem”.

По умолчанию все новые проблемы классифицируются как проблемы-источники. Некоторые проблемы можно вручную переклассифицировать как симптомы проблемы-источника. Подробнее см. cause and symptom events.

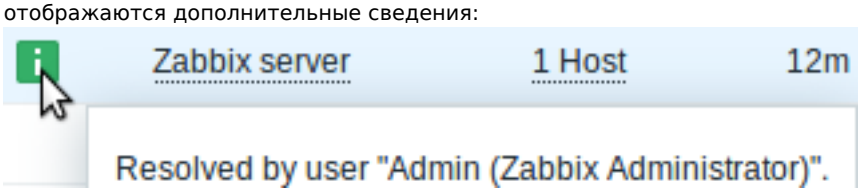
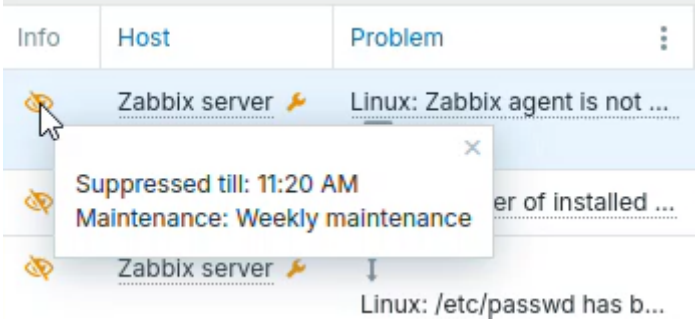


Подробности о проблемах отображаются в настраиваемой таблице. Кнопка *Customize table* в заголовке таблицы позволяет показывать, скрывать или изменять порядок столбцов, а также сбрасывать макет к значениям по умолчанию. Также доступны следующие параметры:

- *Compact view* - уменьшает высоту строк, чтобы на экране помещалось больше проблем.
- *Highlight full row* - использует цвет серьезности проблемы для выделения нерешенных проблем (недоступно в темах с высокой контрастностью).

Вы также можете изменять ширину каждого столбца, перетаскивая его правую границу; двойной щелчок по границе сбрасывает ее. Все настройки применяются сразу и всегда запоминаются для текущего пользователя в активной вкладке filter (по умолчанию или сохраненной).

Доступны следующие столбцы:

Column	Description
Checkbox	Отображаются флажки для выбора проблем. Значки рядом с флажками имеют следующее значение:  - количество событий-симптомов для проблемы-источника;  - развернуть, чтобы показать события-симптомы;  - свернуть, чтобы скрыть события-симптомы;  - это событие-симптом.
Time	Отображается время начала проблемы. Нажатие кнопки *Options* в заголовке столбца открывает доступ к параметру *Show timeline*, который переключает визуальную временную шкалу и группировку проблем (недоступно, когда включен *Compact view*).
Severity	Отображается серьезность проблемы. Серьезность проблемы изначально основана на серьезности соответствующего триггера проблемы, однако после возникновения события ее можно обновить с помощью экрана *Update problem* screen. Цвет серьезности проблемы используется как цвет фона ячейки в течение времени существования проблемы.
Recovery time	Отображается время устранения проблемы. Этот столбец недоступен, когда параметр *Show* в filter установлен в *Problems*.
Status	Отображается статус проблемы: **Problem** - нерешенная проблема **Resolved** - недавно решенная проблема. Вы можете скрыть недавно решенные проблемы с помощью фильтра. Новые и недавно решенные проблемы мигают в течение 2 минут. Решенные проблемы отображаются в общей сложности 5 минут. Оба значения настраиваются в *Administration > General > Trigger displaying options*. Этот столбец недоступен, когда параметр *Show* в filter установлен в *Problems*.
Info	Если проблема закрыта глобальной корреляцией или вручную при обновлении проблемы, отображается зеленый значок информации. При наведении курсора на значок отображаются дополнительные сведения:  Если отображается подавленная проблема, показывается следующий значок: . Когда событие подавлено обслуживанием узла сети, значок отображается оранжевым: . Если применяются и обслуживание, и ручное подавление, приоритет в цвете значка имеет ручное подавление. При наведении курсора на значок отображаются дополнительные сведения: 
Host	Отображается узел сети проблемы. Нажатие на имя узла сети открывает host menu.

Column	Description
Problem	Отображается имя проблемы. Имя проблемы основано на имени соответствующего триггера проблемы. Макросы в имени триггера подставляются в момент возникновения проблемы, и их значения больше не обновляются. Нажатие на имя проблемы открывает event menu. При наведении на значок  после имени проблемы отображается описание триггера (для тех проблем, у которых оно есть). Нажатие кнопки *Options* в заголовке столбца открывает доступ к двум параметрам: - *Show operational data* - отображает оперативные данные после имени проблемы в скобках (только если поле *Operational data* не пустое в trigger configuration). - *Show trigger expression* - отображает выражение триггера проблемы под именем проблемы (недоступно, когда включен *Compact view*).
Operational data	Отображаются динамические данные проблемы в реальном времени. Содержимое зависит от поля *Operational data* в trigger configuration: если задана строка с макросами, она отображается здесь; если поле оставлено пустым, вместо этого отображаются последние значения всех элементов данных в выражении триггера.
Duration	Отображается длительность проблемы. См. также: Negative problem duration
Update	Нажмите ссылку *Update*, чтобы перейти на экран problem update, где можно выполнить различные действия с проблемой, включая добавление комментария и подтверждение проблемы.
Actions	История действий по проблеме отображается с помощью символических значков:  - проблема подтверждена. Этот значок всегда отображается первым.  - были добавлены комментарии. Также отображается количество комментариев.  - серьезность проблемы была повышена (например, Information > Warning).  - серьезность проблемы была понижена (например, Warning > Information).  - серьезность проблемы была изменена, но затем возвращена к исходному уровню (например, Warning > Information > Warning).  - проблема была подавлена.  - подавление проблемы было снято.  - проблема была подавлена из-за обслуживания узла сети.  - подавление проблемы было снято после обслуживания узла сети.  - были выполнены действия. Также отображается количество действий.  - были выполнены действия, по крайней мере одно из них выполняется. Также отображается количество действий.  - были выполнены действия, по крайней мере одно из них завершилось неудачно. Также отображается количество действий. При наведении курсора на значки отображаются всплывающие окна с подробностями о действии. См. viewing details, чтобы узнать больше о значках, используемых во всплывающем окне для выполненных действий.
Tags	Отображаются tags проблемы. Кроме того, могут отображаться теги из внешней системы обработки заявок (см. параметр *Process tags* при настройке webhooks). При наведении курсора или нажатии на значок с тремя точками после отображаемых тегов открывается всплывающее окно со всеми тегами. Нажатие кнопки *Options* в заголовке столбца позволяет задать количество отображаемых тегов (1-3), выбрать способ отображения имен тегов (полностью, сокращенно или не отображать), задать приоритет отображения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).

Column	Description
Tag value	Отображаются значения tag проблемы. Нажатие кнопки ⋮ *Options* в заголовке столбца позволяет переименовать столбец, выбрать, какие значения тегов отображать, или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Custom text	Отображается пользовательский текст, значения макросов или их комбинация. Нажатие кнопки ⋮ *Options* в заголовке столбца позволяет переименовать столбец, ввести пользовательский текст или макросы, либо дублировать столбец (дублированные столбцы можно переименовать или удалить). Supported macros: {HOST.*} (кроме {HOST.METADATA}), {INVENTORY.*}, {ITEM.*} (кроме {ITEM.STATE.ERROR}), {TRIGGER.ID}, {EVENT.ID} и user macros.

Отрицательная длительность проблемы

На самом деле в некоторых распространенных ситуациях возможна отрицательная длительность проблемы, то есть когда время устранения проблемы раньше времени ее создания, например:

- Если какой-либо узел сети контролируется через прокси и происходит сетевая ошибка, из-за которой некоторое время не поступают данные от прокси, триггер nodata(/host/key) будет сработан сервером. Когда соединение будет восстановлено, сервер получит данные элемента данных от прокси со временем из прошлого. Затем проблема nodata(/host/key) будет устранена, и у нее будет отрицательная длительность проблемы;
- Когда данные элемента данных, устраняющие событие проблемы, отправляются Zabbix sender и содержат метку времени, более раннюю, чем время создания проблемы, также будет отображаться отрицательная длительность проблемы.

Note:
Отрицательная длительность проблемы никоим образом не влияет на расчет SLA или отчет о доступности для конкретного триггера; она не уменьшает и не увеличивает время проблемы.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Массовое обновление* - обновить выбранные проблемы, перейдя на экран обновления проблемы

Чтобы использовать этот вариант, отметьте флажки перед соответствующими проблемами, затем нажмите кнопку *Массовое обновление*.

Кнопки

Кнопка справа предлагает следующую опцию:

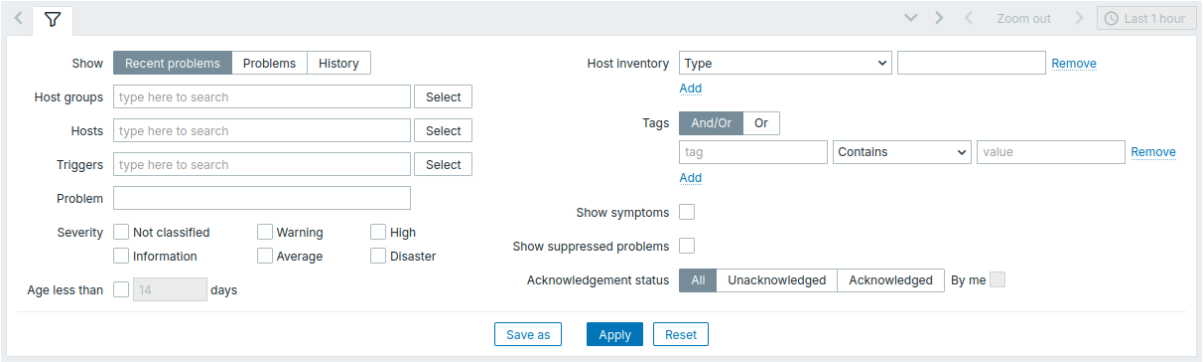
Export to CSV	Экспортировать содержимое со всех страниц в CSV-файл.

Кнопки режима просмотра, общие для всех разделов, описаны на странице Monitoring.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас проблемы. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Фильтр расположен над таблицей. Избранные настройки фильтра можно сохранить в виде вкладок, а затем быстро открывать, нажимая вкладки над фильтром.

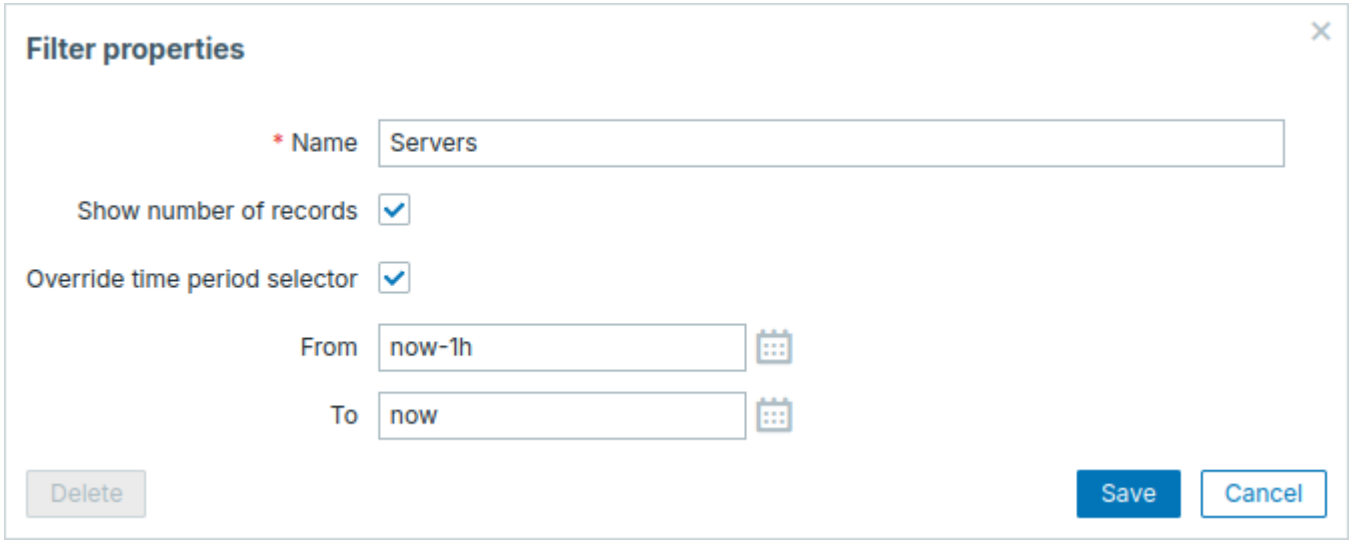


Parameter	Description
Show	Фильтрация по статусу проблемы: **Recent problems** - отображаются нерешенные и недавно решенные проблемы (по умолчанию) **Problems** - отображаются нерешенные проблемы **History** - отображается история всех событий
Host groups	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
Hosts	Фильтрация по одному или нескольким узлам сети.
Triggers	Фильтрация по одному или нескольким триггерам.
Problem	Фильтрация по названию проблемы.
Severity	Фильтрация по серьезности триггера (проблемы).
Age less than	Фильтрация по возрасту проблемы.
Host inventory	Фильтрация по типу и значению инвентарных данных.
Tags	Фильтрация по имени и значению тега события. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия При фильтрации указанные здесь теги будут отображаться первыми вместе с проблемой, если это не переопределено списком *Tag display priority* (см. ниже).
Show symptoms	Установите флажок, чтобы отображать на отдельной строке проблемы, классифицированные как симптомы.
Show suppressed problems	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены и скрыты из-за обслуживания узла сети или подавлены вручную пользователем.
Acknowledgement status	Фильтрация для отображения всех проблем, только неподтвержденных проблем или только подтвержденных проблем. Установите дополнительный флажок, чтобы исключить те проблемы, которые когда-либо были подтверждены вами.

Вкладки для избранных фильтров

Часто используемые наборы параметров фильтра можно сохранять во вкладках.

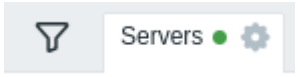
Чтобы сохранить новый набор параметров фильтра, откройте основную вкладку, настройте параметры фильтра, затем нажмите кнопку *Save as*. В новом всплывающем окне задайте *Filter properties*.



Parameter	Description
Name	Имя фильтра, которое будет отображаться в списке вкладок.
Show number of records	Отметьте, если хотите, чтобы рядом с именем вкладки отображалось количество проблем.
Override time period selector	Отметьте, чтобы задать для этого набора фильтров определенный период времени по умолчанию. Если этот параметр включен, изменить период времени для этой вкладки можно будет только путем обновления настроек фильтра. Для вкладок без пользовательского периода времени диапазон можно изменить, нажав кнопку выбора времени в правом верхнем углу (название кнопки зависит от выбранного интервала времени: This week, Last 30 minutes, Yesterday и т. д.). Этот параметр доступен только для фильтров в *Monitoring* > *Problems*.
From/To	Начало и конец периода времени в абсолютном (`Y-m-d H:i:s`) или относительном синтаксисе времени (`now-1d`). Доступно, если установлен флажок *Set custom time period*.

После сохранения фильтр создается как именованная вкладка фильтра и сразу же активируется.

Чтобы изменить свойства фильтра существующего фильтра, нажмите значок шестеренки рядом с именем активной вкладки.



Примечания:

- Чтобы скрыть область фильтра, нажмите на имя текущей вкладки. Чтобы снова открыть область фильтра, еще раз нажмите на имя активной вкладки.
- Поддерживается навигация с клавиатуры: используйте стрелки для переключения между вкладками, нажмите *Enter*, чтобы открыть.
- Кнопки слева/справа над фильтром можно использовать для переключения между сохраненными фильтрами. Кроме того, кнопка со стрелкой вниз открывает выпадающее меню со всеми сохраненными фильтрами, и вы можете нажать нужный.
- Если настройки сохраненного фильтра были изменены (но не сохранены), после имени фильтра отображается зеленая точка. Чтобы обновить фильтр в соответствии с новыми настройками, нажмите кнопку *Update*, которая отображается вместо кнопки *Save as*.
- Макет столбцов таблицы сохраняется отдельно для каждой вкладки фильтра. Чтобы настроить макет сохраненного фильтра, сначала откройте эту вкладку фильтра, а затем настройте столбцы.
- Вкладки фильтров, настройки фильтров и макет столбцов таблицы для каждой вкладки запоминаются в профиле пользователя и восстанавливаются, когда пользователь снова открывает страницу.

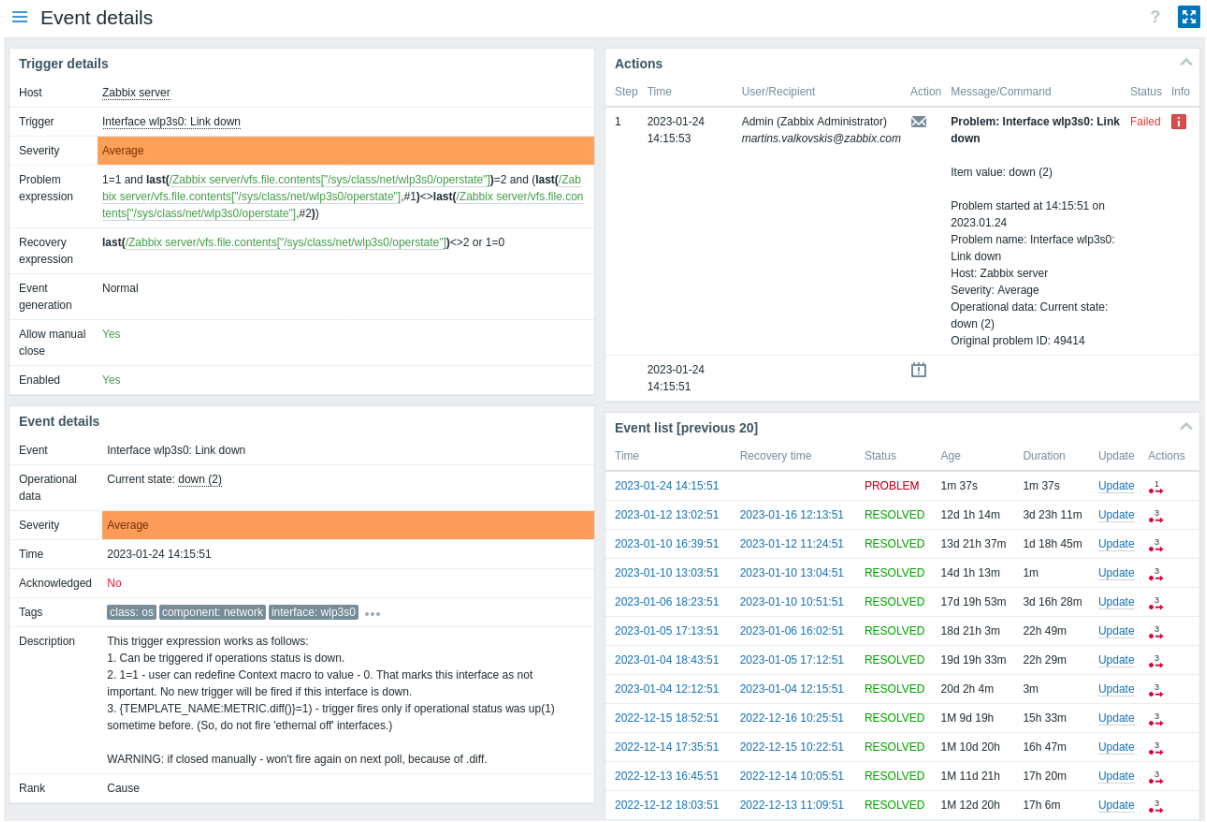
Note:
Чтобы поделиться фильтрами, скопируйте и отправьте другим URL активного фильтра. После открытия этого URL другие пользователи смогут сохранить этот набор параметров как постоянный фильтр в своей учетной записи Zabbix.
 См. также: Параметры страницы.

Кнопки фильтра

Apply	Применить выбранные критерии фильтра (без сохранения).
Reset	Сбросить текущие настройки фильтра и вернуться к сохраненным параметрам текущей вкладки. В главной вкладке при нажатии этой кнопки все поля фильтра вернутся к состоянию по умолчанию.
Save as	Сохранить текущие параметры фильтра в новой вкладке. Доступно только в главной вкладке.
Update	Изменить параметры вкладки на заданные в текущий момент. Недоступно в главной вкладке.

Просмотр подробностей

Время начала проблемы и восстановления в *Monitoring* > *Problems* является ссылками. При нажатии на них открываются дополнительные сведения о событии.



В подробностях события можно увидеть, было ли событие подавлено из-за обслуживания или подавлено вручную пользователем.

Обратите внимание, что серьезность проблемы может отличаться для триггера и события проблемы — если она была изменена для события проблемы с помощью экрана *Update problem* screen.

В списке действий для обозначения типа активности используются следующие значки:

- - событие проблемы сгенерировано.
- - сообщение было отправлено.
- - событие проблемы подтверждено.
- - событие проблемы не подтверждено.
- - добавлен комментарий.
- - серьезность проблемы повышена (например, Information > Warning).

-  - серьезность проблемы понижена (например, Warning > Information).
-  - серьезность проблемы была изменена, но затем возвращена к исходному уровню (например, Warning > Information > Warning).
-  - выполнена удаленная команда.
-  - событие проблемы восстановлено.
-  - проблема закрыта вручную.
-  - проблема подавлена.
-  - подавление проблемы снято.
-  - проблема подавлена из-за обслуживания узла сети.
-  - подавление проблемы снято после обслуживания узла сети.
-  - проблема преобразована в симптомную проблему.
-  - проблема преобразована в проблему-источник.

Причины и проблемы с симптомами

Обзор

По умолчанию все новые проблемы классифицируются как проблемы причины. Некоторые проблемы можно вручную переклассифицировать как проблемы-симптомы проблемы причины.

Например, отключение питания может быть фактической первопричиной того, что какой-то узел сети недоступен или какая-то служба не работает. В этом случае проблемы «узел сети недоступен» и «служба не работает» должны быть классифицированы как проблемы-симптомы «отключения питания» — проблемы причины.

Иерархия причина-симптом поддерживает только два уровня. Проблеме, которая уже является симптомом, нельзя назначать «подчиненные» проблемы-симптомы; любые проблемы, назначенные как симптомы для проблемы-симптома, станут симптомами той же самой проблемы причины.

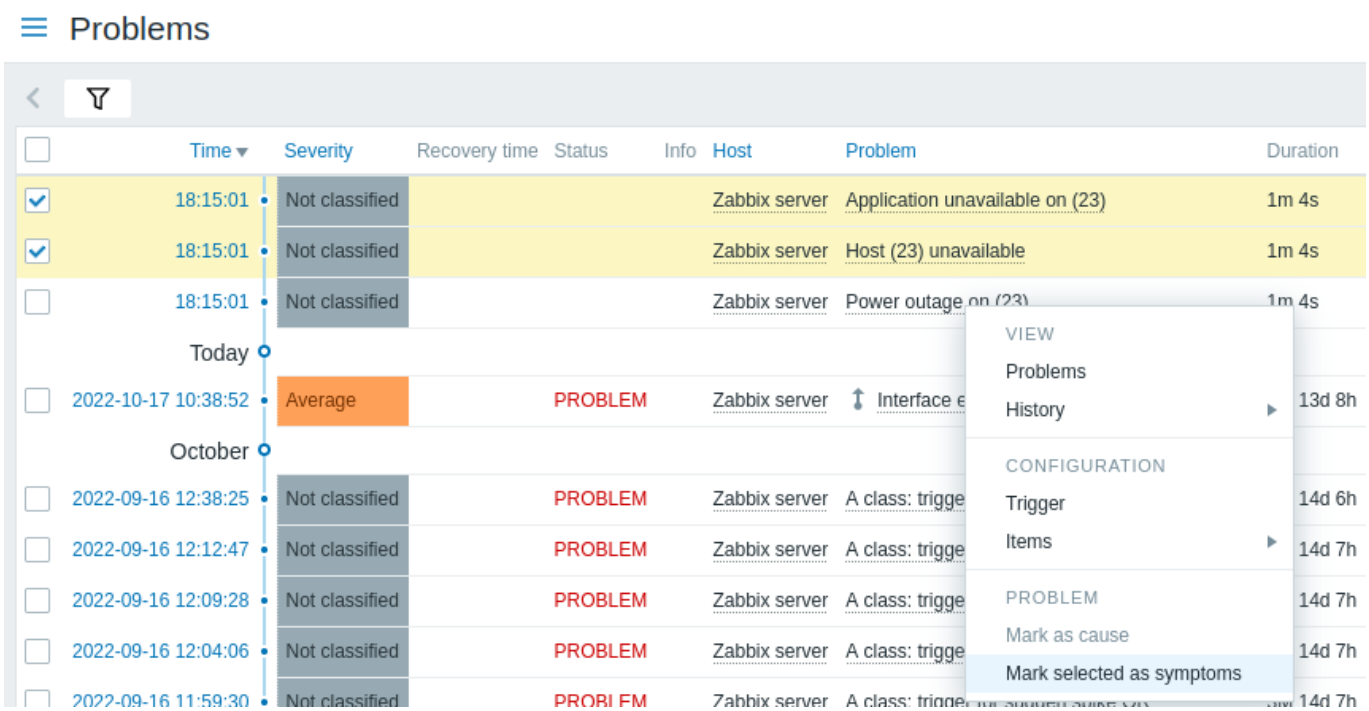
В общих итогах проблем в картах, виджетах веб-интерфейса, таких как *Problems by severity* или *Problem hosts*, и т. д. учитываются только проблемы причины. Однако ранжирование проблем не влияет на службы.

Проблема-симптом может быть связана только с одной проблемой причины. Проблемы-симптомы не разрешаются автоматически, если проблема причины разрешена или закрыта.

Конфигурация

Чтобы переклассифицировать проблему как симптомную проблему, сначала выберите ее в списке проблем. Можно выбрать одну или несколько проблем.

Затем перейдите к проблеме- причине и в ее контекстном меню нажмите пункт *Mark selected as symptoms*.



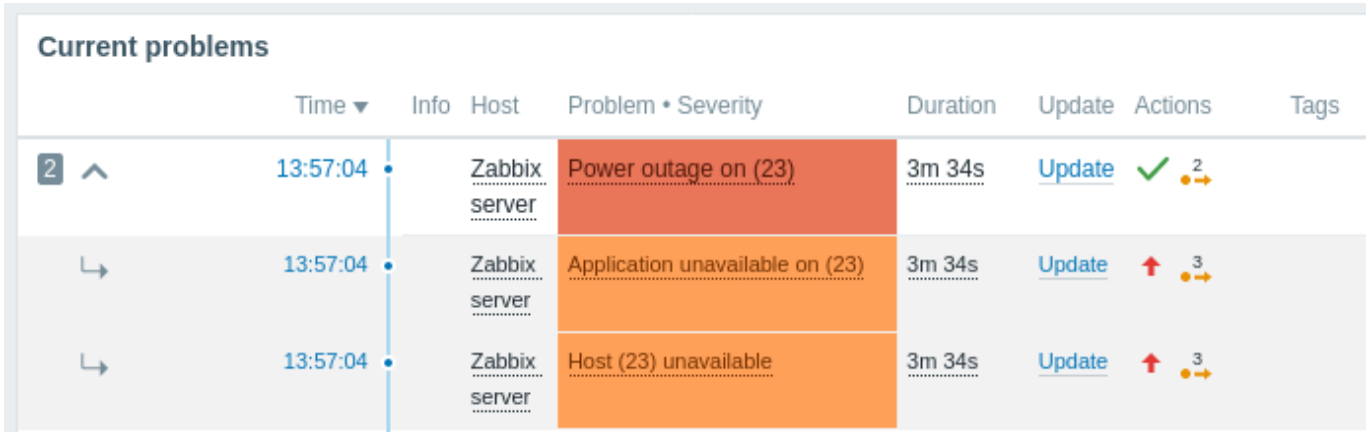
После этого выбранные проблемы будут обновлены сервером и станут симптомными проблемами проблемы- причины.

Пока статус проблемы обновляется, он отображается одним из двух способов:

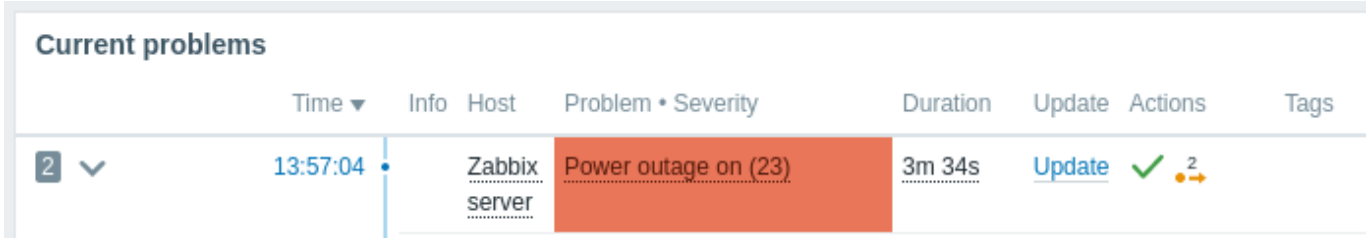
- В столбце Status отображается мигающий статус ”UPDATING”;
- В столбце Info отображается мигающий значок или (это действует, если в фильтре выбраны только *Problems*, и поэтому столбец Status не отображается).

Отображение

Проблемы-симптомы отображаются ниже проблемы-причины и соответствующим образом помечаются в *Monitoring* > *Problems* (а также в виджете панели *Problems*) — значком, меньшим шрифтом и другим фоном.



В свернутом виде видна только проблема-причина; наличие проблем-симптомов обозначается числом в начале строки и значком для разворачивания представления.



Также можно дополнительно отображать проблемы-симптомы обычным шрифтом и на отдельной строке. Для этого выберите *Show symptoms* в настройках фильтра или конфигурации виджета.

Возврат к проблеме-источнику

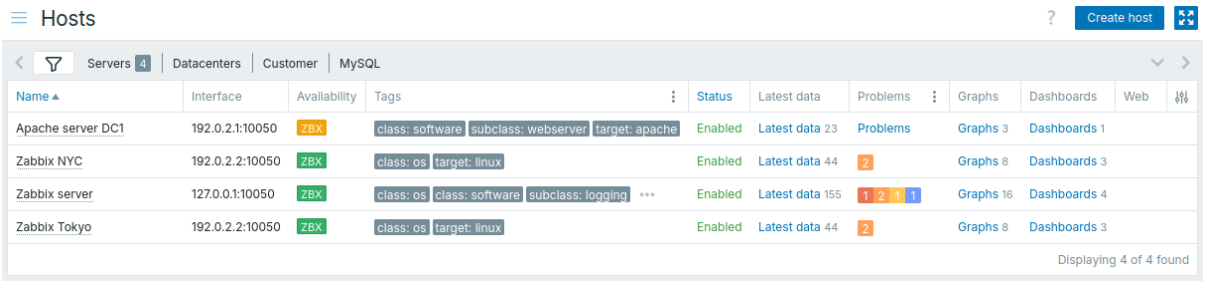
Проблема-симптом может быть возвращена обратно к проблеме-источнику. Для этого выполните одно из следующих действий:

- нажмите пункт *Пометить как причину* в контекстном меню проблемы-симптома;
- отметьте пункт *Преобразовать в причину* на экране обновления проблемы и нажмите *Обновить* (этот вариант также работает, если выбрано несколько проблем).

Узлы сети

Обзор

Раздел *Monitoring* > *Hosts* отображает полный список контролируемых узлов сети с подробной информацией об интерфейсе узла сети, доступности, тегах, текущих проблемах, статусе (включен/выключен) и ссылках для удобной навигации к последним данным узла сети, истории проблем, графикам, панелям и веб-сценариям.



Сведения об узлах сети отображаются в настраиваемой таблице. Кнопка *Customize table* в заголовке таблицы позволяет показывать, скрывать или изменять порядок столбцов, а также сбрасывать макет к значениям по умолчанию. Вы также можете изменять ширину каждого столбца, перетаскивая его правую границу; двойной щелчок по границе сбрасывает ширину. Все изменения применяются немедленно и всегда сохраняются для текущего пользователя в активной вкладке фильтра (по умолчанию или сохраненной).

Доступны следующие столбцы:

Column	Description
Name	Отображаемое имя узла сети. Щелчок по имени открывает меню узла сети. Оранжевый значок гаечного ключа после имени указывает, что этот узел сети находится в обслуживании. Щелкните заголовок столбца, чтобы отсортировать узлы сети по имени по возрастанию или убыванию.
Interface	Отображается основной интерфейс узла сети.

Column	Description
Availability	Отображается доступность узла сети по настроенным интерфейсам. Значки доступности отражают текущее состояние интерфейса узла сети на сервере Zabbix. Поэтому, если вы отключите узел сети во веб-интерфейсе, его доступность обновится после того, как сервер Zabbix синхронизирует изменения конфигурации. Аналогично, если вы включите узел сети, его доступность обновится после того, как сервер Zabbix синхронизирует изменения конфигурации и опросит узел сети. Значки доступности отображают только те типы интерфейсов (Agent, SNMP, IPMI, JMX), которые настроены. При наведении курсора на значок отображается всплывающее окно со списком всех интерфейсов того же типа с подробностями, статусом и ошибками. Для интерфейса Agent во всплывающем окне отображаются интерфейсы (passive) и активные проверки. Если у узла сети настроены только активные проверки, значок интерфейса Agent отображается даже в том случае, если интерфейс Agent не настроен. Для узлов сети без интерфейсов столбец пуст. Статус отдельного интерфейса узла сети определяется соединением между включенным элементом данных, использующим этот интерфейс, и узлом сети. Статус может быть следующим: **Available** - соединение с узлом сети успешно; **Not available** - соединение с узлом сети не удалось (тайм-аут, проблемы с брандмауэром и т. д.); **Unknown** - попытка соединения с узлом сети не выполнялась или результат неизвестен. Дополнительные сведения о том, как сервер Zabbix определяет статус интерфейса, см. в разделах Unknown interface status и Unreachable/unavailable host interface settings. Статус всех интерфейсов одного типа у узла сети (Agent, SNMP, IPMI, JMX) определяется теми интерфейсами, которые используются как минимум одним включенным элементом данных. Статус обозначается цветом значка: **Green** - все интерфейсы доступны; **Yellow** - как минимум один интерфейс недоступен, и как минимум один доступен или неизвестен; **Red** - все интерфейсы недоступны; **Gray** - как минимум один интерфейс неизвестен, но ни один не является недоступным. **Доступность активных проверок.** Если на узле сети включена хотя бы одна активная проверка, активные проверки также влияют на общую доступность интерфейса Agent, как описано выше. Для определения доступности активных проверок сообщения heartbeat отправляются в потоке активных проверок агента. Частота сообщений heartbeat управляется параметром `HeartbeatFrequency` в конфигурации Zabbix агента или agent 2 (по умолчанию 60 секунд, диапазон 0-3600). Активные проверки считаются недоступными, если heartbeat активной проверки старше чем 2 x HeartbeatFrequency секунд. **Примечание:** агенты Zabbix версии ниже 6.2.x не отправляют heartbeat активных проверок, поэтому доступность их узлов сети остается неизвестной.
Tags	Отображаются теги узла сети. При наведении курсора или щелчке по значку с тремя точками после отображаемых тегов открывается всплывающее окно со всеми тегами. Щелчок по кнопке *Options* в заголовке столбца позволяет задать количество отображаемых тегов (1–3), выбрать способ отображения имен тегов (полный, сокращенный или без отображения), установить приоритет отображения тегов или дублировать столбец (дублированные столбцы можно переименовывать или удалять).
Tag value	Отображаются значения тегов узла сети. Щелчок по кнопке *Options* в заголовке столбца позволяет переименовать столбец, выбрать, какие значения тегов отображать, или дублировать столбец (дублированные столбцы можно переименовывать или удалять).
Status	Статус узла сети - *Enabled* или *Disabled*. Щелкните заголовок столбца, чтобы отсортировать узлы сети по статусу по возрастанию или убыванию.

Column	Description
Latest data	Щелчок по ссылке откроет раздел *Monitoring* > *Latest data* со всеми последними данными, собранными с узла сети. Количество элементов данных с последними данными отображается серым цветом.
Problems	Отображается количество открытых проблем для узла сети, отсортированных по серьезности. Цвет каждого значка проблемы показывает серьезность, а число - количество проблем для этой серьезности. Щелчок по значку открывает *Monitoring* > *Problems* для этого узла сети; если у узла сети нет проблем, вместо этого отображается ссылка *Problems*. Щелчок по кнопке *Options* в заголовке столбца дает доступ к опции *Show suppressed problems*, которая включает проблемы, которые в противном случае были бы подавлены (скрыты) из-за обслуживания узла сети или отдельного подавления проблемы.
Graphs	Щелчок по ссылке отобразит графики, настроенные для узла сети. Количество графиков отображается серым цветом. Если у узла сети нет графиков, ссылка не отображается.
Dashboards	Щелчок по ссылке отобразит панели, настроенные для узла сети. Количество панелей отображается серым цветом. Если у узла сети нет панелей, ссылка не отображается.
Web	Щелчок по ссылке отобразит веб-сценарии, настроенные для узла сети. Количество веб-сценариев отображается серым цветом. Если у узла сети нет веб-сценариев, ссылка не отображается.
Custom text	Отображается пользовательский текст, значения макросов или их сочетание. Щелчок по кнопке *Options* в заголовке столбца позволяет переименовать столбец, ввести пользовательский текст или макросы, либо дублировать столбец (дублированные столбцы можно переименовывать или удалять). Поддерживаемые макросы: `{HOST.*}` (кроме `{HOST.METADATA}`), `{INVENTORY.*}` и пользовательские макросы.

Кнопки

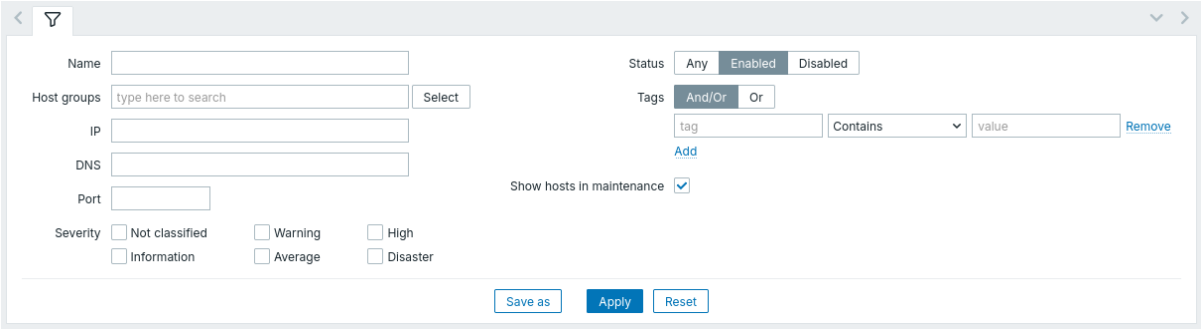
Создать узел сети позволяет создать новый узел сети. Эта кнопка доступна только пользователям с ролями «Администратор» и «Суперадминистратор».

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас узлы сети. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Фильтр расположен над таблицей. Можно фильтровать узлы сети по имени, группе узлов сети, IP или DNS, порту интерфейса, тегам, серьезности проблемы, статусу (включен/выключен/любой); также можно выбрать, отображать ли узлы сети, которые в данный момент находятся на обслуживании.



Parameter	Description
Name	Фильтрация по видимому имени узла сети.
Host groups	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
IP	Фильтрация по IP-адресу.
DNS	Фильтрация по DNS-имени.
Port	Фильтрация по номеру порта.

Parameter	Description
Severity	Фильтрация по серьезности проблемы. По умолчанию отображаются проблемы всех уровней серьезности. Проблемы отображаются, если они не подавлены.
Status	Фильтрация по статусу узла сети.
Tags	Фильтрация по имени и значению тега узла сети. Узлы сети можно фильтровать по тегам уровня узла сети, а также по тегам всех связанных шаблонов, включая вложенные шаблоны. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия
Show hosts in maintenance	Установите флажок, чтобы отображать узлы сети, находящиеся на обслуживании (по умолчанию отображаются).

Сохранение фильтра

Настройки избранного фильтра можно сохранить в виде вкладок, а затем быстро открывать, нажимая соответствующую вкладку над фильтром.

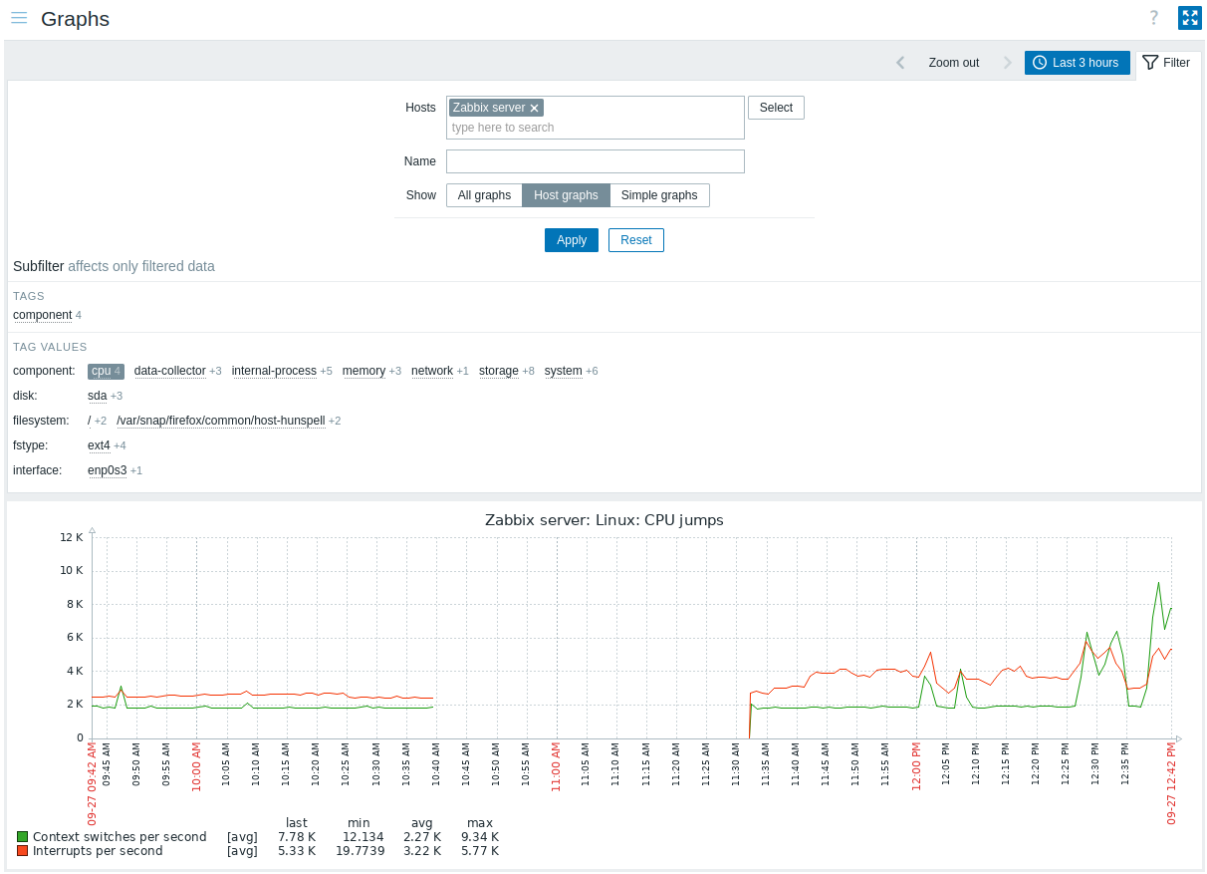
Подробнее о сохранении фильтров.

Графики

Обзор

К графикам узла сети можно перейти из *Monitoring* > *Hosts*, щелкнув Graphs для соответствующего узла сети.

Можно отображать любой пользовательский график, настроенный для узла сети, а также любой простой график.



Графики сортируются по:

- имени графика (пользовательские графики)
- имени элемента данных (простые графики)

Графики для отключенных узлов сети также доступны.

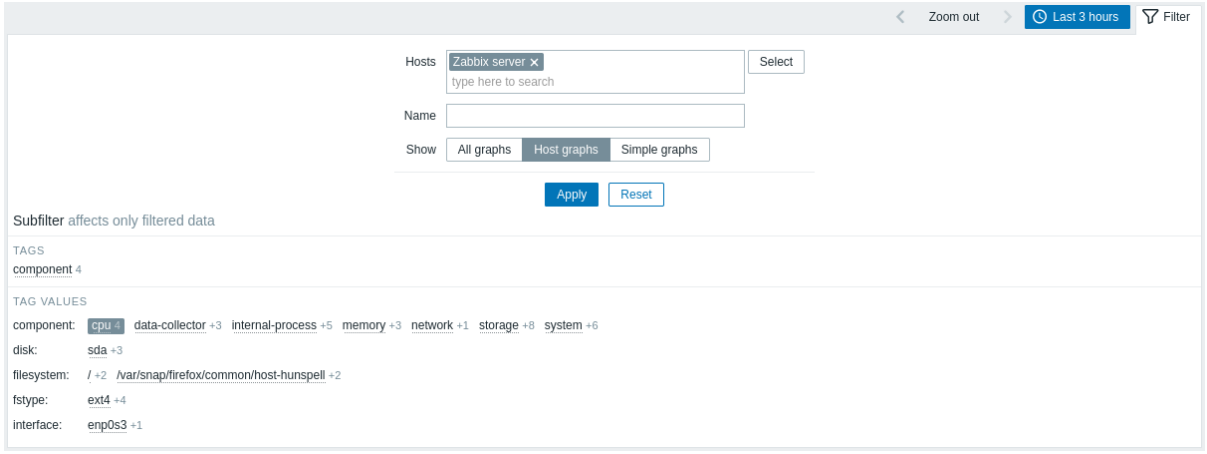
Выбор периода времени

Обратите внимание на выбор периода времени над графиком. Он позволяет выбирать часто используемые периоды одним щелчком мыши.

См. также: *Выбор периода времени*

Использование фильтра

Чтобы просмотреть определённый график, выберите его в фильтре. Фильтр позволяет указать узел сети, имя графика и параметр *Показать* (все графики/графики узла сети/простые графики).



Если в фильтре не выбран узел сети, графики не отображаются.

Использование подфильтра

Подфильтр позволяет дополнительно изменить фильтрацию, примененную основным фильтром.

Он содержит кликабельные ссылки для быстрого доступа к связанным графикам. Графики связаны общей сущностью — именем или значением тега. При нажатии на имя или значение тега оно выделяется серым фоном, а графики немедленно фильтруются (нет необходимости нажимать *Применить* в основном фильтре). Нажатие на другое имя или значение тега добавляет его к результатам фильтрации. Повторное нажатие на имя или значение тега отменяет фильтрацию по нему.

Подфильтры создаются на основе отфильтрованных данных, количество которых ограничено 1000 записями. Если вы хотите видеть в подфильтре больше записей, увеличьте значение параметра *Ограничение результатов поиска и фильтрации* (в разделе *Администрирование* > *Общие* > *Веб-интерфейс*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы всегда отображать актуальную информацию о доступных параметрах фильтрации и их количестве.

Количество отображаемых сущностей по горизонтали ограничено 100. Если сущностей больше, в конце отображается значок с тремя точками; он не является кликабельным. Вертикальные списки (например, теги с их значениями) ограничены 20 элементами. Если элементов больше, отображается значок с тремя точками; он не является кликабельным.

Число рядом с каждой кликабельной сущностью указывает количество графиков, содержащих ее в результатах основного фильтра.

После выбора одной сущности рядом с другими доступными сущностями отображаются числа со знаком «плюс», указывающие, сколько графиков можно добавить к текущему выбору.

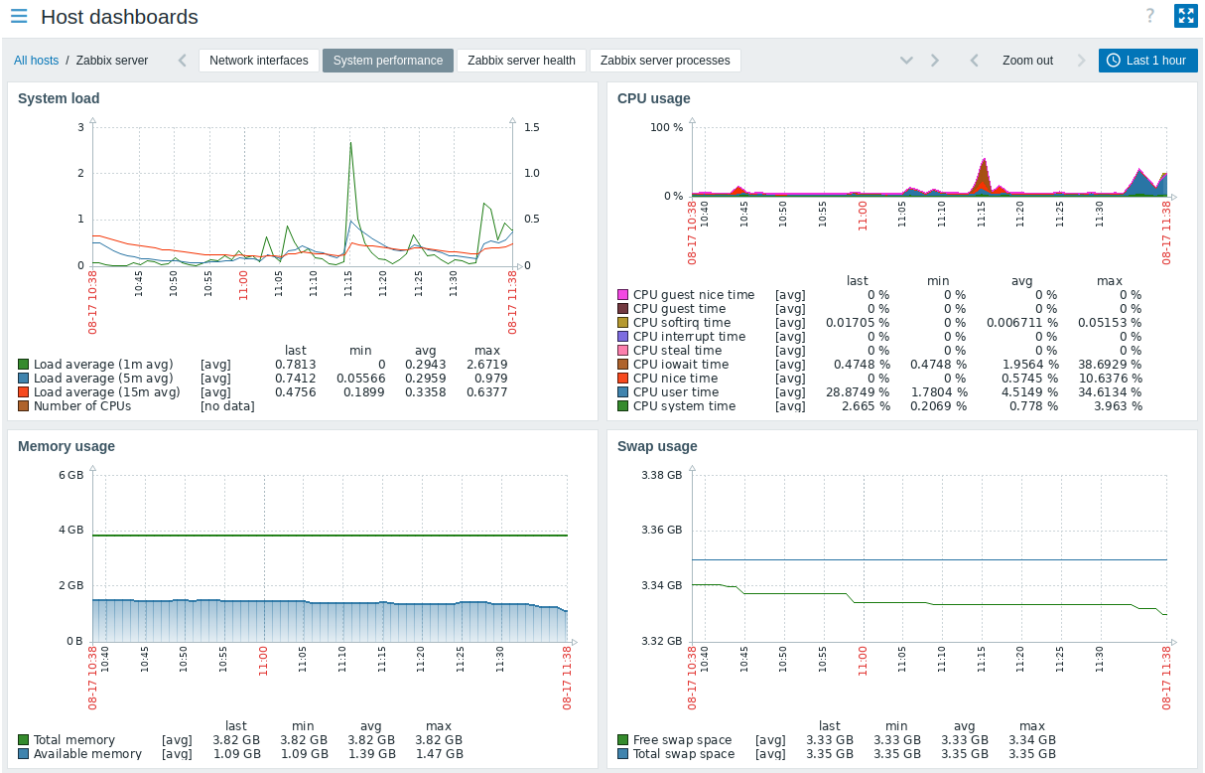
Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Панели узла сети

Обзор

Панели узла сети выглядят аналогично глобальным панелям, однако панели узла сети не имеют владельца и отображают данные только для выбранного узла сети.



При просмотре панелей узла сети можно переключаться между настроенными панелями, нажимая:

- вкладки панелей;
- кнопки со стрелками под заголовком;
- кнопку со стрелкой под заголовком, которая отображает полный список доступных панелей узла сети.

Чтобы перейти в раздел *Мониторинг* > *Узлы сети*, нажмите ссылку навигации *Все узлы сети* под заголовком в левом верхнем углу.

Конфигурация

Панели узла сети настраиваются на уровне шаблона. После привязки шаблона к узлу сети для этого узла сети создаются панели узла сети. Обратите внимание, что панели узла сети *нельзя* настраивать в разделе *Панели*, который предназначен для глобальных панелей.

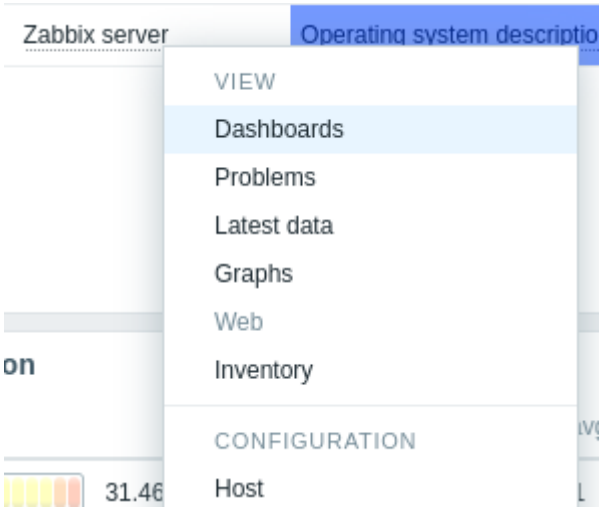
Виджеты панелей узла сети также можно настраивать только на уровне шаблона, за исключением изменения интервала обновления. Кроме того, виджеты панелей узла сети можно копировать только в другие панели узла сети в пределах того же шаблона. Обратите внимание, что виджеты из глобальных панелей *нельзя* копировать в панели узла сети.

Note:
До Zabbix 5.2 панели узла сети назывались экранами узла сети. При импорте более старого шаблона, содержащего экраны, импорт экранов будет проигнорирован.

Доступ

К панелям узла сети можно получить доступ:

- после поиска имени узла сети в глобальном поиске (нажмите ссылку *Dashboards* в результатах поиска);
- после нажатия на имя узла сети в *Инвентарь* > *Узлы сети* (нажмите ссылку *Dashboards* в обзоре узла сети);
- из меню узла сети, нажав *Dashboards*.

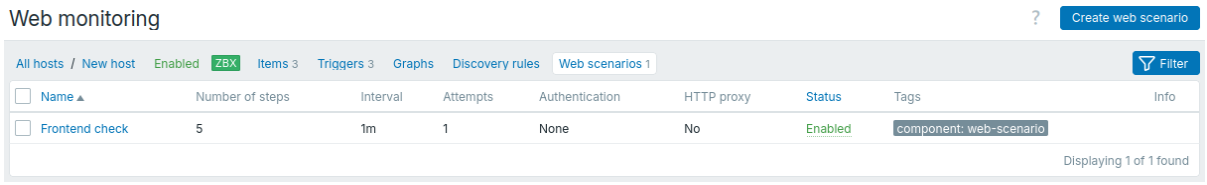


Обратите внимание, что к панелям узла сети *нельзя* получить прямой доступ в разделе *Dashboards*, который предназначен для глобальных панелей.

Веб-сценарии

Обзор

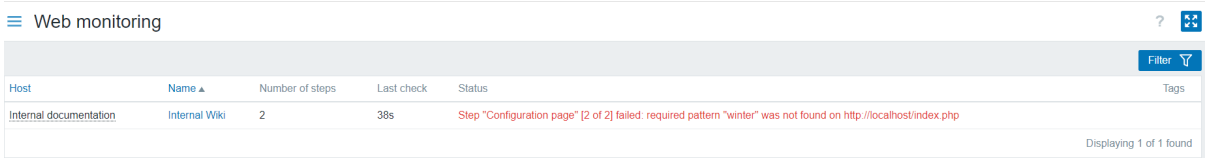
Информацию о web scenario узла сети можно получить в *Мониторинг* > *Узлы сети*, нажав Web для соответствующего узла сети.



Нажатие на имя узла сети открывает меню узла сети. Данные отключенных узлов сети также доступны. Имя отключенного узла сети отображается красным цветом.

Статус web scenario может быть следующим:

- OK (зеленый) - все шаги успешно выполнены, и Zabbix собрал данные для сценария.
- Error (красный) - один или несколько шагов завершились с ошибкой (см. сообщение об ошибке, чтобы понять, какие параметры могут требовать исправления).
- Unknown (серый) - статус шагов еще недоступен или данные еще не были собраны.



Максимальное количество сценариев, отображаемых на странице, зависит от настройки профиля пользователя *Rows per page* setting.

По умолчанию отображаются только значения, попадающие в последние 24 часа. Это ограничение введено для улучшения времени первоначальной загрузки больших страниц с последними данными. Вы можете увеличить этот период, изменив значение параметра *Max history display period* в разделе меню *Administration > General > GUI*.

Имя сценария является ссылкой на более подробную статистику по нему:



Использование фильтра

На странице отображается список всех веб-сценариев выбранного узла сети. Чтобы просмотреть веб-сценарии для другого узла сети или группы узлов сети, не возвращаясь на страницу *Monitoring > Hosts*, выберите этот узел сети или группу в фильтре. Вы также можете фильтровать сценарии по тегам.

Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Monitoring.

Последние данные

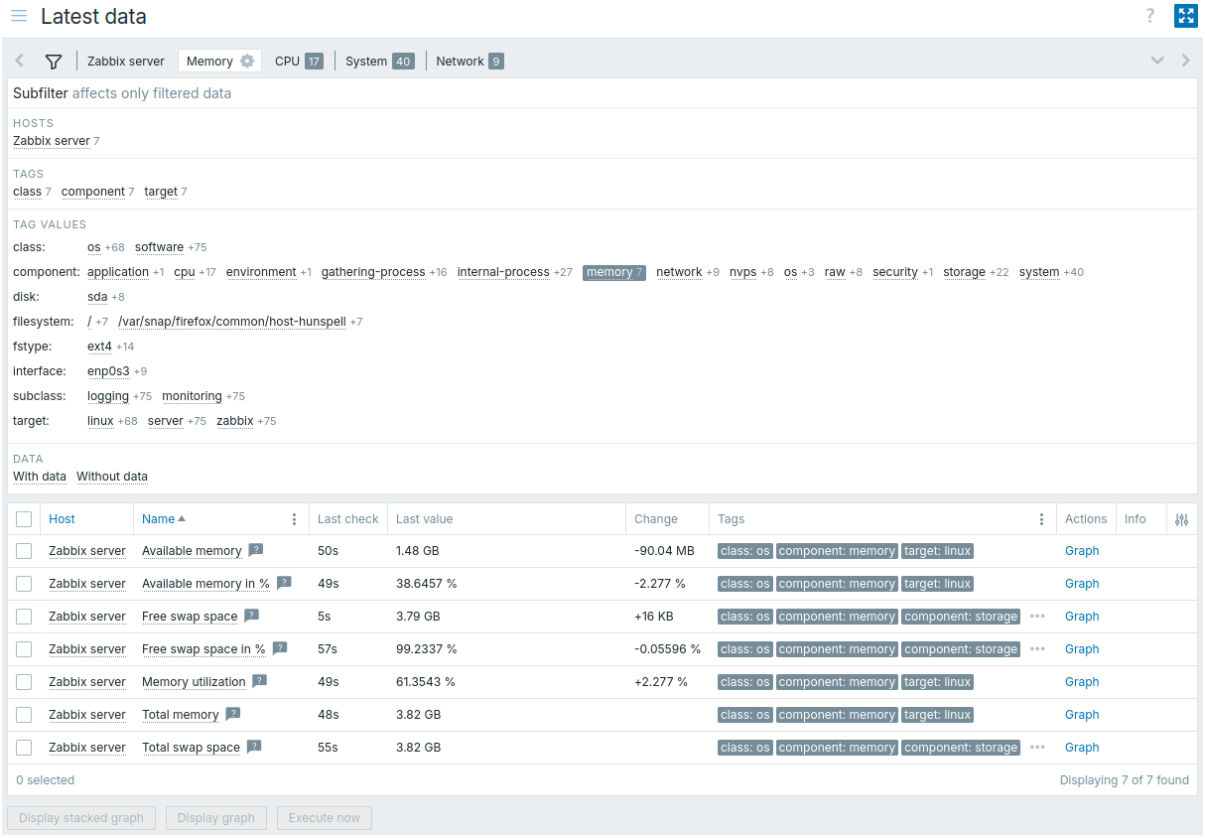
Обзор

Раздел *Monitoring > Latest data* отображает последние значения, собранные элементами данных.

Этот раздел содержит следующие элементы:

- Фильтр
- Подфильтр
- Список элементов данных

Note:
Подфильтр и список элементов данных отображаются только если фильтр задан и есть результаты для отображения.



Подробности о последних значениях отображаются в настраиваемой таблице. Кнопка *Customize table* в заголовке таблицы позволяет показывать, скрывать или изменять порядок столбцов, а также сбрасывать макет к значениям по умолчанию. Вы также можете изменять ширину каждого столбца, перетаскивая его правую границу; двойной щелчок по границе сбрасывает ее. Все изменения применяются сразу и всегда сохраняются для текущего пользователя во вкладке активного фильтра (по умолчанию или сохраненной).

Доступны следующие столбцы:

Column	Description
Host	Имя узла сети, к которому относится элемент данных. Щелчок по имени открывает контекстное меню узла сети. Если узел сети находится в обслуживании, после имени узла сети отображается оранжевый значок гаечного ключа . Если узел сети отключен, его имя отображается красным цветом. Обратите внимание, что данные отключенных узлов сети (включая графики и списки значений элементов данных) доступны в разделе *Latest data*.
Name	Имя элемента данных. Щелчок по имени открывает меню элемента данных. Рядом с именем элемента данных для всех элементов, у которых есть описание, отображается значок вопросительного знака . Наведите указатель на значок, чтобы отобразить всплывающую подсказку с описанием элемента данных. Щелчок по кнопке *Options* в заголовке столбца дает доступ к параметру *Show item key*, который отображает ключ элемента данных под именем каждого элемента данных.
Interval	Интервал обновления элемента данных.
History	Период хранения истории элемента данных.
Trends	Период хранения трендов элемента данных.
Type	Тип элемента данных.

Column	Description
Last check	Время, прошедшее с момента последней проверки элемента данных.
Last value	Последнее значение элемента данных. Значения отображаются с учетом преобразования единиц измерения и сопоставления значений. Наведите указатель на значение, чтобы отобразить необработанные данные (обрезанные до 8192 символов). Двоичные значения отображаются как заполнитель (*binary value*) вместо фактического значения. По умолчанию отображаются только значения, полученные за последние 24 часа. Это ограничение улучшает время начальной загрузки для больших страниц с последними данными; чтобы его изменить, обновите значение параметра *Max history display period* в *Administration* > *General* > *GUI*.
Change	Разница между предыдущим значением и последним значением. Для элементов данных с частотой обновления 1 день и более изменение никогда не отображается (при настройках по умолчанию). В этом случае последнее значение вообще не будет отображаться, если оно было получено более 24 часов назад.
Tags	Теги, связанные с элементом данных. Щелчок по тегу включает его в подфильтре, тем самым фильтруя список элементов данных по этому тегу (и любым другим тегам, выбранным в подфильтре). После щелчка такие теги больше не являются кликабельными; чтобы изменить фильтрацию по тегам, необходимо использовать подфильтр. Щелчок по кнопке *Options* в заголовке столбца позволяет задать количество отображаемых тегов (1-3), выбрать способ отображения имен тегов (полный, сокращенный или без имен), задать приоритет отображения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Tag value	Отображаются значения тега элемента данных. Щелчок по кнопке *Options* в заголовке столбца позволяет переименовать столбец, выбрать, какие значения тегов отображать, или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Actions	Ссылка на простой график/историю значений элемента данных.
Info	Дополнительная информация об элементе данных. Если у элемента данных есть ошибки (например, он стал неподдерживаемым), отображается информационный значок. Наведите указатель на значок, чтобы увидеть подробности.
Custom text	Отображается пользовательский текст, значения макросов или их комбинация. Щелчок по кнопке *Options* в заголовке столбца позволяет переименовать столбец, ввести пользовательский текст или макросы, либо дублировать столбец (дублированные столбцы можно переименовать или удалить). Поддерживаемые макросы: {HOST.*} (кроме {HOST.METADATA}), {INVENTORY.*}, {ITEM.*} (кроме {ITEM.STATE.ERROR}) и пользовательские макросы.

Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Массовые действия

Кнопки под списком предлагают массовые действия с одним или несколькими выбранными элементами:

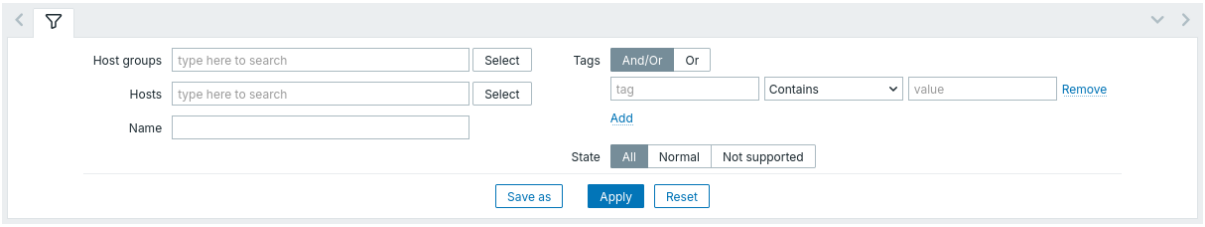
- *Показать составной график* - показать составной график ad-hoc.
- *Показать график* - показать простой график ad-hoc.
- *Выполнить сейчас* - немедленно выполнить проверку новых значений элемента данных. Поддерживается только для **пассивных** проверок (см. подробности). Эта опция доступна только для узлов сети с доступом на чтение и запись. Доступ к этой опции для узлов сети с правами только на чтение зависит от параметра роли пользователя под названием *Вызывать "Выполнить сейчас" на узлах сети только для чтения*.

Чтобы использовать эти опции, установите флажки напротив соответствующих элементов данных, затем нажмите нужную кнопку.

Использование фильтра

С помощью фильтра можно отображать только интересующие вас элементы данных. Для повышения производительности поиска данные ищутся без раскрытия макросов.

Значок фильтра  расположен над списком элементов данных и подфильтром. Нажмите его, чтобы развернуть фильтр.



Фильтр позволяет сузить список элементов данных по группе узлов сети, узлу сети, имени элемента данных, тегам и состоянию. Указание родительской группы узлов сети в фильтре автоматически выбирает все вложенные группы узлов сети. Подробнее о фильтрации по тегам см. в разделе *Мониторинг* > *Проблемы*.

Параметр фильтра *Показать подробности* позволяет расширить отображаемую информацию об элементах данных. Установите его, чтобы отображать такие сведения, как интервал обновления элемента данных, настройки хранения истории и трендов, тип элемента данных и ошибки элемента данных (корректный/неподдерживаемый).

Сохранение фильтра

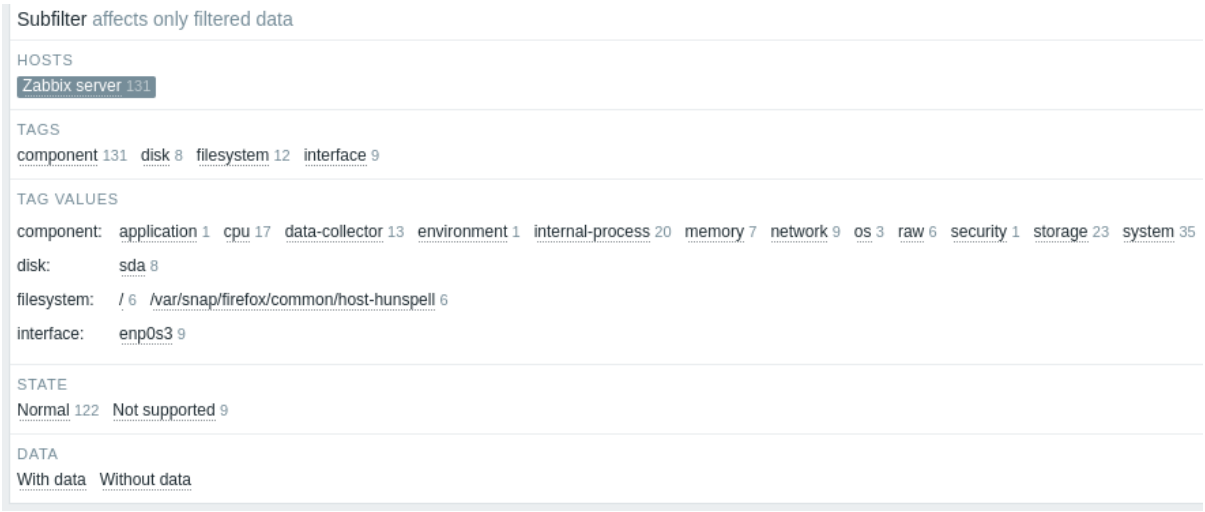
Настройки избранного фильтра можно сохранить в виде вкладок, а затем быстро открывать, щелкнув соответствующую вкладку над фильтром.

Подробнее см. в разделе сохранение фильтров.

Использование подфильтра

Подфильтр позволяет дополнительно изменить фильтрацию, заданную в основном фильтре.

Он содержит ссылки, по которым можно щелкнуть для быстрого доступа к связанным элементам данных. Элементы данных связаны общей сущностью: узлом сети, именем или значением тега, состоянием элемента данных или состоянием данных. При выборе сущности она выделяется серым фоном, а элементы данных немедленно фильтруются (нет необходимости нажимать *Применить* в основном фильтре). Выбор другой сущности добавляет ее к результатам фильтрации. Повторный выбор сущности отменяет фильтрацию по ней.



Подфильтры создаются на основе отфильтрованных данных, количество которых ограничено 1000 записями. Если имеется 20 узлов сети, каждый из которых содержит 100 элементов данных (всего 2000 записей), в подфильтре будут отображаться только половина узлов сети. Чтобы отображать в подфильтре больше записей, необходимо увеличить значение параметра *Ограничение результатов поиска и фильтрации* (в разделе *Администрирование* > *Общие* > *Графический интерфейс*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы информация о доступных параметрах фильтрации и их счетчиках всегда оставалась актуальной.

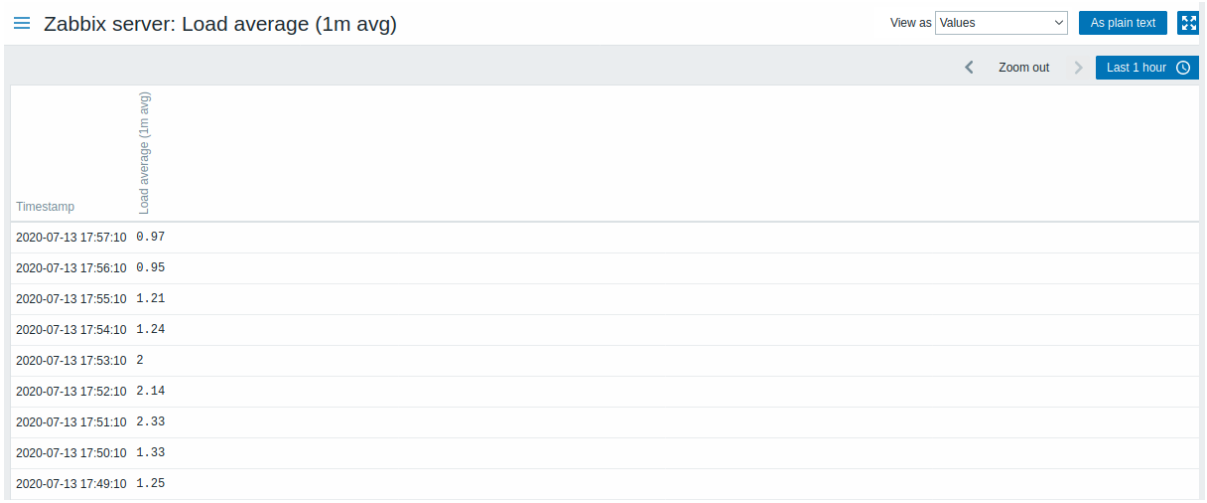
Для каждой группы сущностей (узлы сети, теги, значения тегов и т. д.) отображается до 10 строк сущностей. Если сущностей больше, список можно развернуть, чтобы отобразить не более 1000 записей (значение SUBFILTER_VALUES_PER_GROUP в определениях веб-интерфейса), нажав значок с тремя точками ••• в конце списка. Для *Значений тегов* список можно развернуть, чтобы отобразить не более 200 имен тегов с соответствующими значениями. Обратите внимание, что после полного разворачивания список нельзя свернуть.

Число рядом с каждой доступной для выбора сущностью указывает количество сгруппированных в ней элементов данных (на основе результатов основного фильтра). После выбора сущности рядом с другими доступными сущностями отображаются числа со знаком «плюс», указывающие, сколько элементов данных можно добавить к текущему выбору. Сущности, не содержащие элементов данных, не отображаются, если только ранее не были выбраны в подфильтре.

Графики и история

Столбец *Graph/History* в списке элементов данных предлагает следующие ссылки:

- **History** - для всех текстовых элементов, ведет к спискам (*Values/500 latest values*), отображающим историю предыдущих значений элемента данных.
- **Graph** - для всех числовых элементов, ведет к простому графику. Обратите внимание, что при отображении графика выпадающий список в правом верхнем углу также предлагает возможность переключиться на *Values/500 latest values*.



Значения, отображаемые в этом списке, являются необработанными, то есть постобработка не применяется.

Note:
Общее количество отображаемых значений определяется значением параметра *Limit for search and filter results*, установленного в *Administration* > *General* > *GUI*.

Карты

Обзор

В разделе *Monitoring* > *Maps* вы можете настраивать, управлять и просматривать карты сети.

Когда вы открываете этот раздел, вы увидите либо последнюю карту, к которой вы обращались, либо список всех карт, к которым у вас есть доступ.

Все карты могут быть общедоступными или частными. Общедоступные карты доступны всем пользователям, тогда как частные карты доступны только их владельцу и пользователям, с которыми карта была расшарена.

Список карт



Отображаемые данные:

Column	Description
Name	Имя карты. Щелкните имя, чтобы просмотреть карту.
Width	Отображается ширина карты.
Height	Отображается высота карты.

Column	Description
Actions	Доступны два действия: **Properties** - задать общие свойства карты **Edit** - открыть сетку для добавления элементов карты

Чтобы настроить новую карту, нажмите кнопку *Create map* в правом верхнем углу. Чтобы импортировать карту из файла YAML, XML или JSON, нажмите кнопку *Import* в правом верхнем углу. Пользователь, который импортирует карту, будет назначен ее владельцем.

Две кнопки под списком предлагают несколько вариантов массового редактирования:

- *Export* - экспортировать карты в файл YAML, XML или JSON
- *Delete* - удалить карты

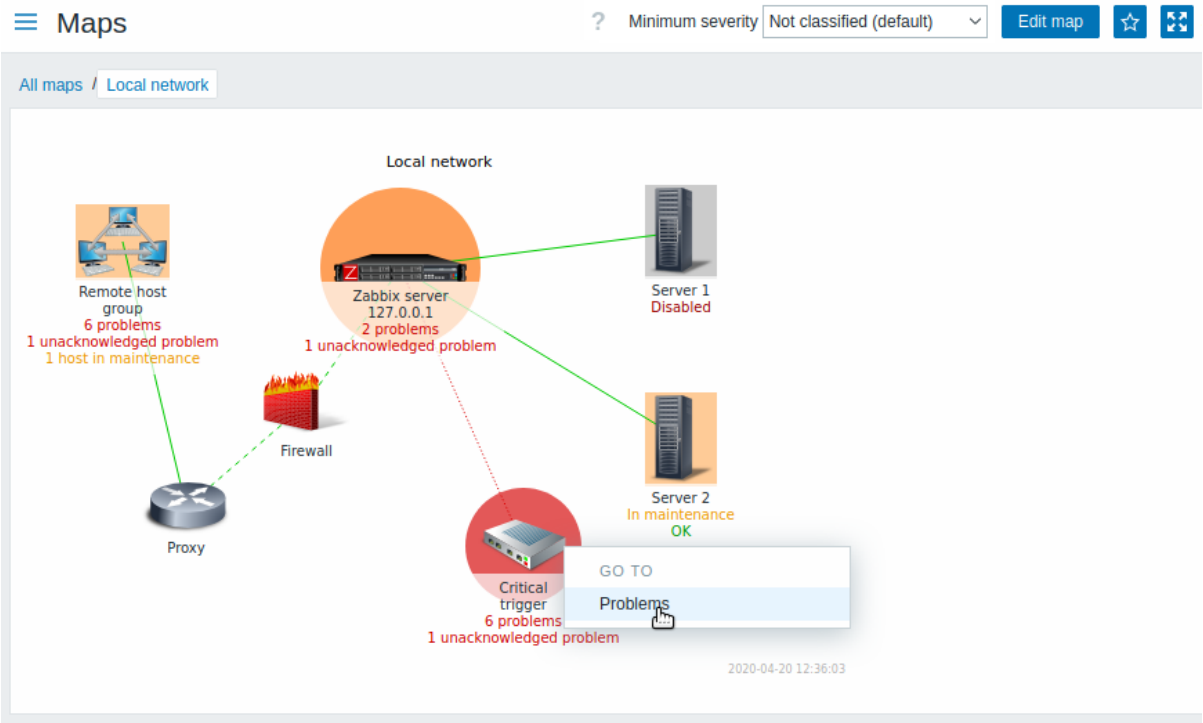
Чтобы использовать эти параметры, отметьте флажки напротив соответствующих карт, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те карты, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Просмотр карт

Чтобы просмотреть карту, щелкните ее имя в списке всех карт.



Вы можете использовать выпадающий список в строке заголовка карты, чтобы выбрать минимальный уровень серьезности триггеров проблем для отображения. Серьезность, отмеченная как *default*, — это уровень, заданный в конфигурации карты. Если карта содержит под-карту, при переходе к под-карте будет сохранен уровень серьезности карты более высокого уровня (кроме случая, если это *Not classified*; в этом случае он не будет передан в под-карту).

Выделение значков

Если элемент карты находится в состоянии проблемы, он выделяется круглым контуром. Цвет заливки круга соответствует цвету уровня серьезности проблемы. Вместе с элементом отображаются только проблемы с выбранным уровнем серьезности или выше. Если все проблемы подтверждены, вокруг круга отображается толстая зеленая граница.

Кроме того:

- узел сети в режиме обслуживания выделяется оранжевым закрашенным квадратом. Обратите внимание, что выделение режима обслуживания имеет приоритет над выделением по уровню серьезности проблемы, если элементом карты является узел сети.
- отключенный (не отслеживаемый) узел сети выделяется серым закрашенным квадратом.

Выделение отображается, если в конфигурации карты установлен флажок *Выделение значков*.

Маркеры недавних изменений

Красные треугольники, направленные внутрь элемента, указывают на недавнее изменение состояния триггера — произошедшее в течение последних 30 минут. Эти треугольники отображаются, если в разделе настройки карты установлен флажок *Отмечать элементы при изменении состояния триггера*.

Ссылки

Щелчок по элементу карты открывает меню с доступными ссылками. Щелчок по имени узла сети открывает меню узла сети.

Кнопки

Кнопки справа предлагают следующие варианты:

	Перейти к редактированию содержимого карты.
	Добавить карту в виджет избранного в Панелях.
	Карта находится в виджете избранного в Панелях. Нажмите, чтобы удалить карту из виджета избранного.

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Удобное для чтения описание на картах

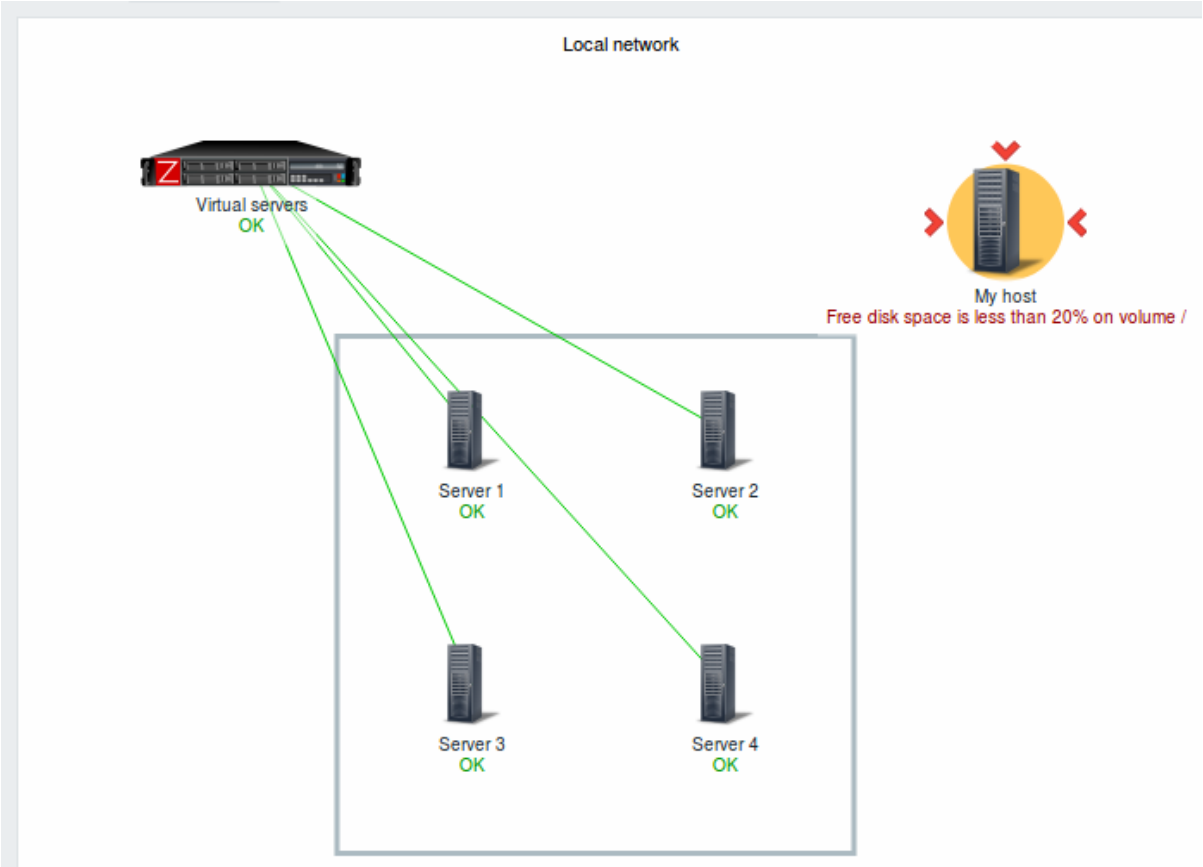
Доступно скрытое свойство `aria-label`, позволяющее просматривать информацию о карте с помощью программы чтения с экрана. Доступны общее описание карты и описание отдельных элементов в следующем формате:

- для описания карты: < >, <* * >, <* >.
- для описания одного элемента с одной проблемой: < >, < >, < >, < >.
- для описания одного элемента с несколькими проблемами: < >, < >, < >, <* >.
- для описания одного элемента без проблем: < >, < >, < >.

Например, доступно следующее описание:

' , 1 6 , 1 . , : , , 20%

для следующей карты:



Ссылка на карту сети

На карты сети можно ссылаться с помощью GET-параметров `sysmapid` и `mapname`. Например,

`http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network`

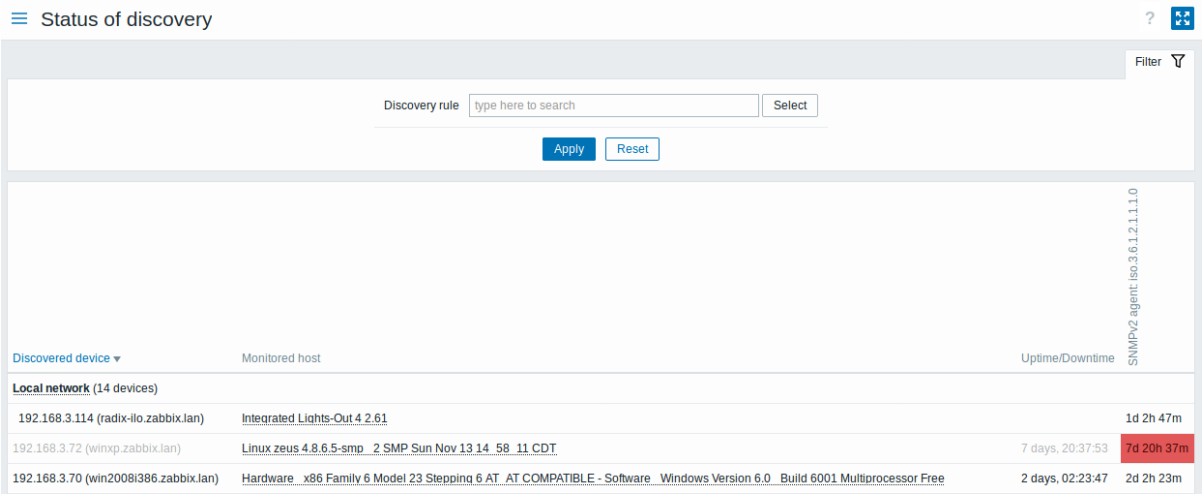
откроет карту с таким именем (Local network).

Если указаны и `sysmapid` (ID карты), и `mapname` (имя карты), то `mapname` имеет более высокий приоритет.

Обнаружение

Обзор

В разделе *Мониторинг* > *Обнаружение* отображаются результаты сетевого обнаружения. Обнаруженные устройства отсортированы по правилу обнаружения.



Отображаемые данные:

Столбец	Описание
Обнаруженное устройство	Обнаруженные устройства перечислены и сгруппированы по правилу обнаружения. При нажатии на правило обнаружения открывается меню правила со ссылкой на форму настройки правила обнаружения. Если у устройства несколько IP-адресов (согласно *Критерию уникальности устройства* правила обнаружения), они перечисляются вместе: сначала основной IP-адрес устройства, затем остальные IP-адреса (с отступом). Основным IP-адресом является наименьший IP-адрес, для которого в столбце *Контролируемый узел сети* указан узел сети, или, если такого адреса нет, наименьший IP-адрес устройства. Устройства можно сортировать по основному IP-адресу в порядке возрастания или убывания; остальные IP-адреса устройства всегда перечисляются под его основным IP-адресом в порядке возрастания.
Контролируемый узел сети	Если устройство уже контролируется, в этом столбце отображается имя узла сети. При нажатии на имя узла сети открывается меню узла сети.
Время работы/простоя	В этом столбце отображается продолжительность обнаружения устройства или его отсутствия после предыдущего обнаружения. Для устройства с несколькими IP-адресами продолжительность отображается только для его основного IP-адреса.
Проверка обнаружения	Отображается состояние отдельной службы (проверки обнаружения) для каждого обнаруженного устройства. Красная ячейка означает, что служба недоступна. В ячейке также указывается время работы или простоя службы. Этот столбец отображается только в том случае, если служба обнаружена хотя бы на одном обнаруженном устройстве.

Note:
Количество правил обнаружения и общее количество обнаруженных устройств, отображаемых в этом разделе, ограничено параметром *Ограничение результатов поиска и фильтрации* в разделе *Администрирование > Общее > Веб-интерфейс*. Правила или устройства, превышающие это ограничение, не отображаются, при этом никакого уведомления об этом нет.

Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Monitoring.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила обнаружения, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Если в фильтре ничего не выбрано, отображаются все включенные правила обнаружения. Чтобы выбрать конкретное правило обнаружения для отображения, начните вводить его имя в фильтре. Будут перечислены все подходящие включенные правила обнаружения, доступные для выбора. Можно выбрать более одного правила обнаружения.

APM

Обзор

Меню *APM* предназначено для функций мониторинга производительности приложений (APM) в Zabbix. В нем отображаются данные телеметрии, которые ваши приложения и службы отправляют с использованием OpenTelemetry — открытого стандарта для сбора данных телеметрии. Эти данные помогают увидеть, как запросы проходят через приложения и службы, измерить их производительность и определить причину медленных или неудачных операций.

Меню содержит следующие разделы, соответствующие типам сигналов OpenTelemetry:

- Трассировки
- Метрики
- Журналы

В отличие от раздела *Мониторинг > Последние данные*, в котором отображаются значения, собранные элементами данных Zabbix, в разделах *APM* отображаются данные телеметрии непосредственно из источника данных, поэтому создавать элементы данных не требуется. Чтобы отображать данные в разделах *APM*:

1. Укажите источник данных телеметрии в разделе *Администрирование > Источник данных*.

2. Убедитесь, что данные телеметрии записываются в источник данных прокси Zabbix с включенным APM или сборщиком OpenTelemetry. Обратите внимание, что сервер Zabbix не может получать данные телеметрии; это может делать только прокси Zabbix.

Чтобы использовать данные телеметрии в триггерах или графиках, используйте элемент данных запрос телеметрии.

Трассировки

Обзор

Раздел *APM* > *Трассировки* отображает трассировки, собранные из ваших приложений и сервисов.

Трассировка представляет собой путь, который запрос проходит через приложения и сервисы. Например, когда пользователь нажимает кнопку *Оформить заказ* в интернет-магазине, запрос может пройти от сервиса веб-интерфейса через сервисы оформления и оплаты до базы данных.

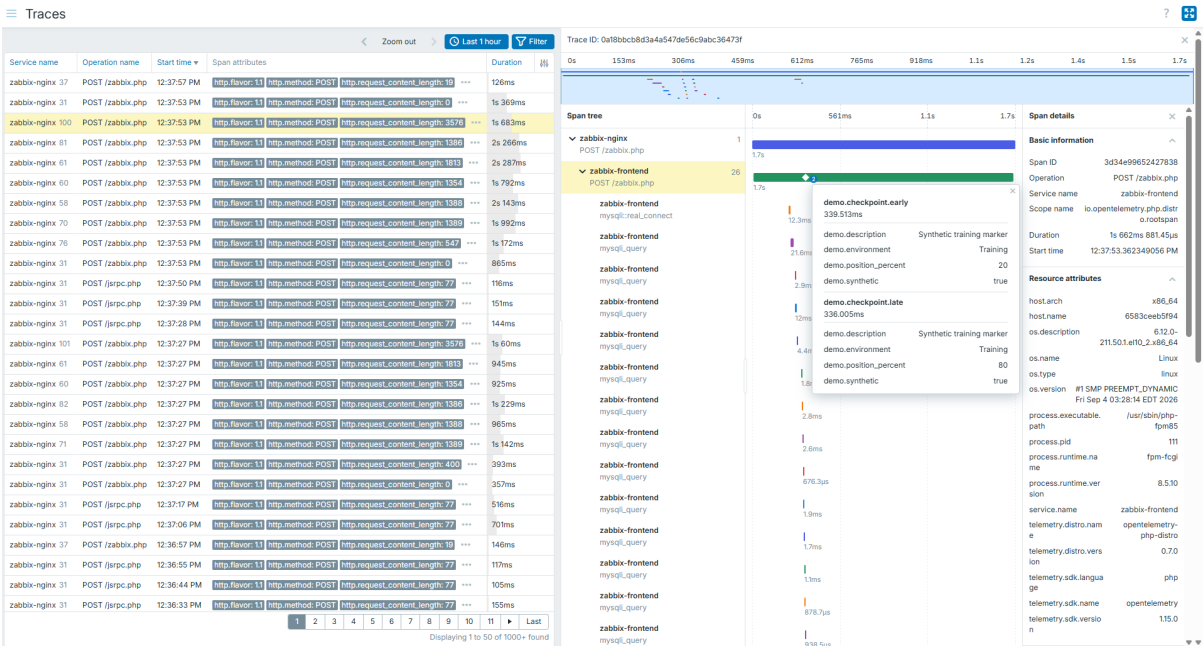
Трассировка состоит из **интервалов**. Каждый интервал представляет одну операцию, например HTTP-запрос, который сервис веб-интерфейса отправляет сервису оформления заказа, запрос к базе данных для сохранения заказа или вызов функции, рассчитывающей стоимость доставки.

Интервалы образуют иерархию. **Корневой интервал** представляет первую операцию запроса. Каждый другой интервал является **дочерним интервалом** интервала, во время которого была запущена его операция. Например, получение сервисом веб-интерфейса запроса *Оформить заказ* является корневым интервалом, а обработка заказа сервисом оформления — его дочерним интервалом.

Каждый интервал создается **сервисом** (например, `zabbix-frontend`; имя определяется атрибутом ресурса OpenTelemetry `service.name`) и областью инструментирования — частью кода, которая его создала (например, библиотекой, отслеживающей HTTP-запросы).

Чтобы предоставить контекст для каждой операции, интервалы также содержат ключ-значение **атрибуты**: атрибуты интервала описывают саму операцию (например, `http.method: POST`), а атрибуты ресурса описывают сервис и среду, в которой он работает (например, `host.name: eabf8c2a5ddc`).

Дополнительные сведения о трассировках см. в документации OpenTelemetry.

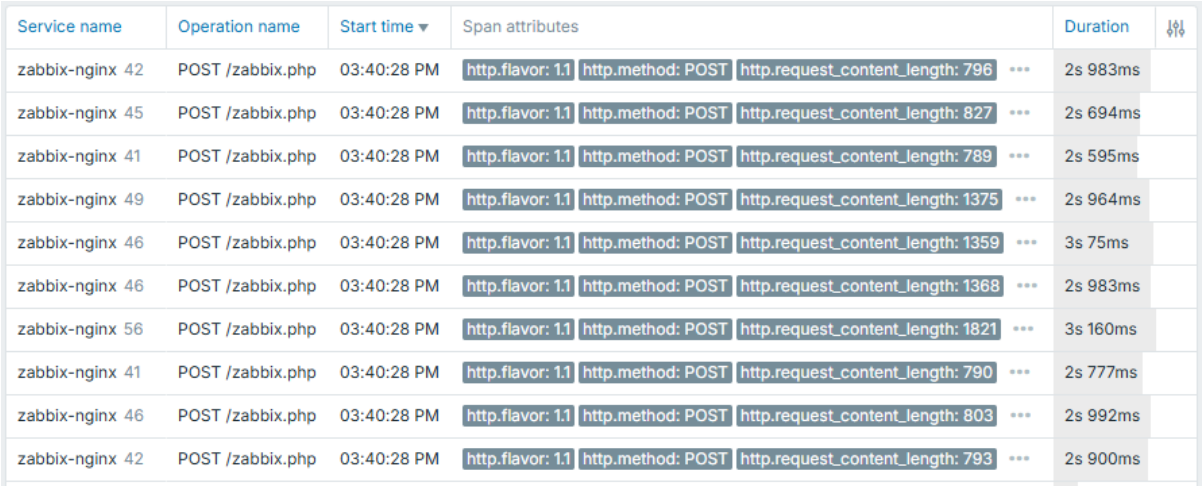


Раздел *APM* > *Трассировки* содержит следующие элементы:

- Кнопка режима киоска — отображать только содержимое страницы без меню и заголовка.
- Выбор периода времени — выбрать период времени, за который необходимо отображать трассировки.
- Фильтр — отображать только интересующие вас трассировки.
- Список трассировок — просматривать трассировки, описанные их корневыми интервалами.
- Панель трассировки — просматривать все интервалы выбранной трассировки на временной шкале.

Список трассировок

Каждая строка в списке трассировок представляет трассировку; в столбцах отображаются данные её корневого спана.



Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796 •••	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827 •••	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789 •••	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375 •••	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359 •••	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368 •••	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821 •••	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790 •••	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803 •••	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793 •••	2s 900ms

Трассировки отображаются в настраиваемой таблице. Кнопка *Настроить таблицу* в заголовке таблицы позволяет отображать, скрывать или изменять порядок столбцов, а также восстанавливать макет по умолчанию. Размер каждого столбца также можно изменить, перетащив его правую границу; для сброса размера дважды щёлкните границу. Все изменения применяются немедленно и сохраняются для текущего пользователя.

Список трассировок может содержать следующие столбцы:

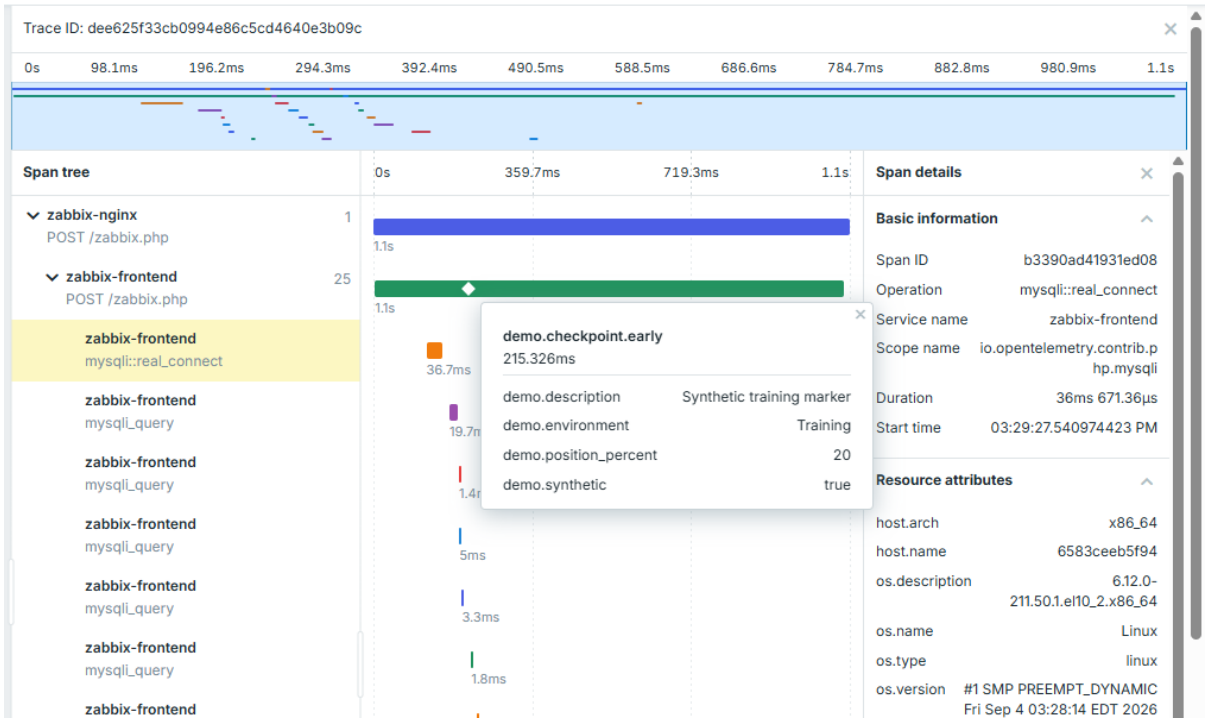
Note:
Каждый столбец соответствует столбцу в таблице `otel_traces` источника данных телеметрии; список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии. Большинство столбцов имеют те же названия, что и в источнике данных (например, *Идентификатор трассировки* соответствует столбцу `TraceId`). Если названия отличаются, столбец источника данных указан в таблице ниже.

Столбец	Описание
Идентификатор трассировки	Идентификатор трассировки, общий для всех спанов, относящихся к трассировке.
Идентификатор спана	Идентификатор корневого спана.
Состояние трассировки	Дополнительные данные, если они есть, которые инструменты трассировки добавляют к трассировке (см. W3C Trace Context).
Имя сервиса	Сервис, создавший корневой спан, за которым следует количество спанов в трассировке. Если какие-либо спаны имеют код состояния *Ошибка* (см. ниже), красный значок показывает их количество.
Имя операции	Операция, которую представляет корневой спан (например, `POST /zabbix.php`). Столбец источника данных: `SpanName`.
Тип спана	Роль корневого спана: **Не указано** - не указан. **Внутренний** - операция, выполняемая внутри приложения. **Сервер** - обработка входящего запроса. **Клиент** - исходящий запрос. **Производитель** - создание асинхронной операции (например, отправка сообщения). **Потребитель** - обработка асинхронной операции.
Имя области	Область инструментирования, создавшая корневой спан.
Версия области	Версия области инструментирования.
Код состояния	Состояние операции, которую представляет корневой спан: **Не задано** - корневому спану не присвоено явное состояние. **OK** - операция успешно завершена. **Ошибка** - операция завершена с ошибкой.
Сообщение о состоянии	Дополнительная информация, если она есть, о состоянии.
Время начала	Дата и время начала корневого спана. Столбец источника данных: `Timestamp`.
Атрибуты спана	Атрибуты корневого спана. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.
Атрибуты ресурса	Атрибуты ресурса корневого спана. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.

Столбец	Описание
Длительность	Длительность трассировки; серая полоса показывает её относительно других трассировок.

Панель трассировки

Нажмите на трассировку в списке трассировок, чтобы открыть панель трассировки, в которой на временной шкале отображаются все спаны трассировки: какие операции выполнялись, в каком порядке и сколько времени заняла каждая из них.

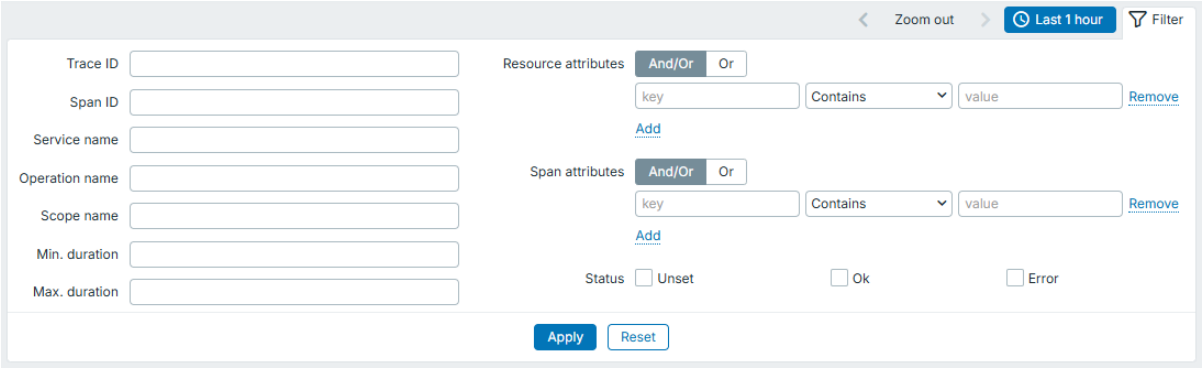


Панель трассировки может содержать следующие элементы:

Элемент	Описание
Идентификатор трассировки	Идентификатор выбранной трассировки.
Обзор временной шкалы трассировки	Вся трассировка со спанами, расположенными максимум в 8 строках (несколько спанов могут находиться в одной строке). Перетаскивайте границы выбранного диапазона или сам диапазон, чтобы увеличить масштаб временной шкалы спанов для этой части трассировки.
Дерево спанов	Спаны в иерархическом порядке; для каждого отображаются его сервис (полужирным шрифтом), имя операции и количество непосредственных дочерних спанов.
Временная шкала спанов	Спаны в виде полос: положение и длина показывают, когда начался каждый спан и сколько времени он выполнялся. Маркеры обозначают события (например, исключения); наведите указатель на маркер или нажмите его, чтобы просмотреть подробности.
Сведения о спане	Нажмите на спан в дереве спанов, чтобы просмотреть его сведения и атрибуты.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас трассировки.



Чтобы отфильтровать данные по атрибутам, введите ключ атрибута, выберите оператор и введите значение атрибута.

Поддерживаются следующие операторы:

- **Содержит** - *(по умолчанию)* значение атрибута содержит введенную строку.
- **Равно** - значение атрибута равно введенной строке.
- **Не содержит** - значение атрибута не содержит введенную строку.
- **Не равно** - значение атрибута не равно введенной строке.
- **Существует** - атрибут с введенным ключом существует.
- **Не существует** - атрибут с введенным ключом не существует.

Если вы добавили несколько условий, выберите способ их оценки:

- **И/ИЛИ** - *(по умолчанию)* должны выполняться все условия; условия с одинаковым ключом атрибута объединяются с помощью *ИЛИ*.
- **ИЛИ** - должно выполняться хотя бы одно условие.

Метрики

Обзор

В разделе *APM* > *Метрики* отображаются точки данных метрик, собранные из ваших приложений и служб.

Метрика — это измерение (например, длительность запроса, количество обработанных заказов или использование памяти). Приложения отправляют свои значения в виде **точек данных**, содержащих значение (или агрегированные значения), измеренное в определённый момент времени. Метрика похожа на элемент данных Zabbix, а точка данных — на значение этого элемента данных.

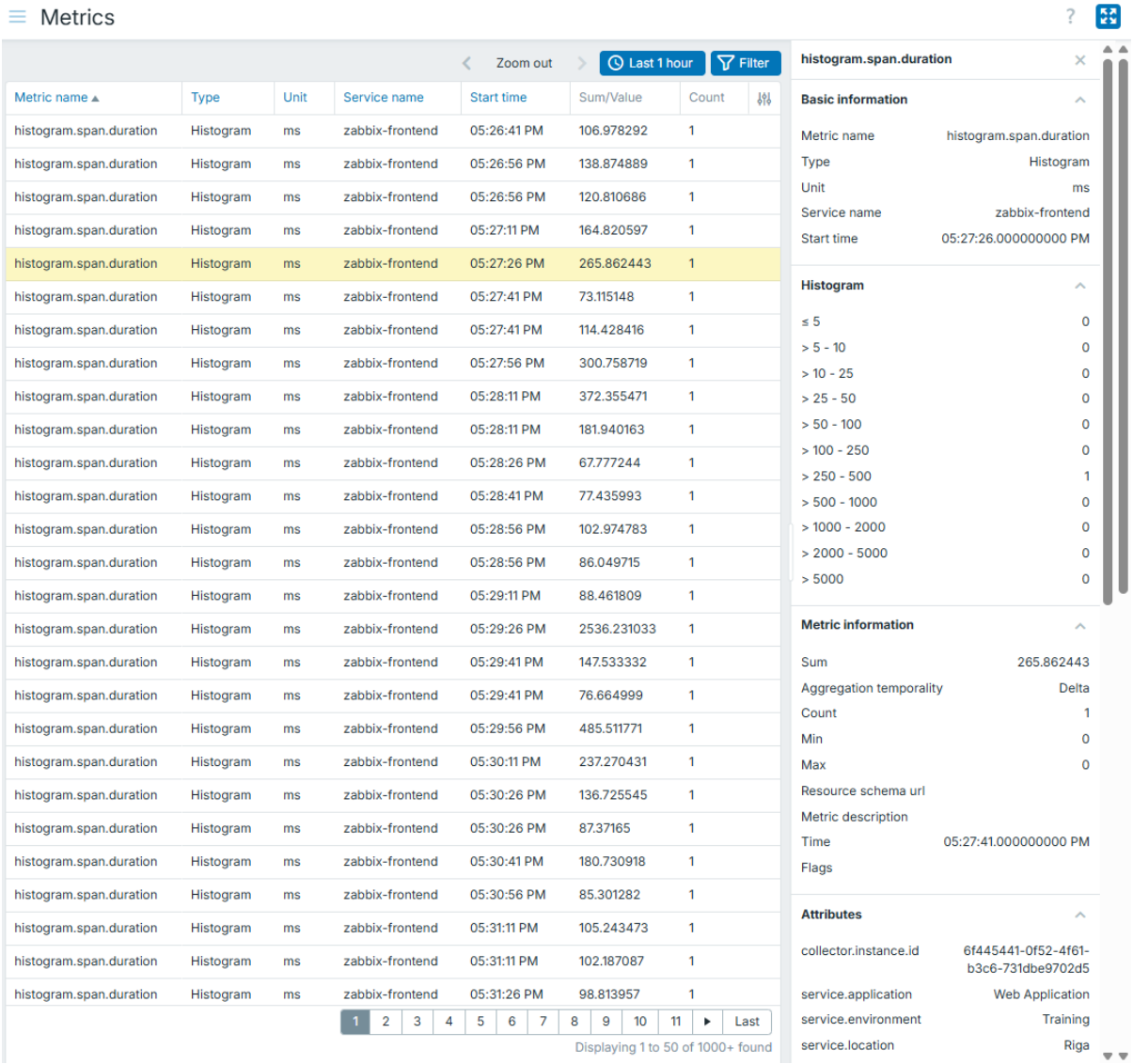
Точка данных может иметь один из следующих типов:

- **Gauge** — значение, измеренное в определённый момент времени (например, текущее использование памяти).
- **Sum** — сумма измерений за определённый период времени (например, общее количество запросов).
- **Histogram** — количество измерений за определённый период времени, попавших в каждый из нескольких фиксированных диапазонов значений, называемых корзинами (например, сколько запросов выполнялось от 0 до 100 мс, от 100 до 250 мс или от 250 до 500 мс).
- **Exponential histogram** — аналог гистограммы, но каждая корзина шире предыдущей в одинаковое количество раз (например, 1–2 мс, 2–4 мс, 4–8 мс).

Каждая точка данных создаётся **службой** (например, `zabbix-frontend`; имя задаётся атрибутом ресурса OpenTelemetry `service.name`) и областью инструментирования — частью кода, которая её создала (например, библиотекой, измеряющей длительность HTTP-запросов).

Чтобы предоставить контекст для каждого измерения, точки данных также содержат ключ-значение **атрибуты**: атрибуты точек данных описывают, что было измерено, и позволяют различать точки данных одной и той же метрики (например, `http.request.method: POST`), а атрибуты ресурсов описывают службу и среду, в которой она работает (например, `host.name: eabf8c2a5ddc`).

Дополнительную информацию о метриках см. в документации OpenTelemetry.

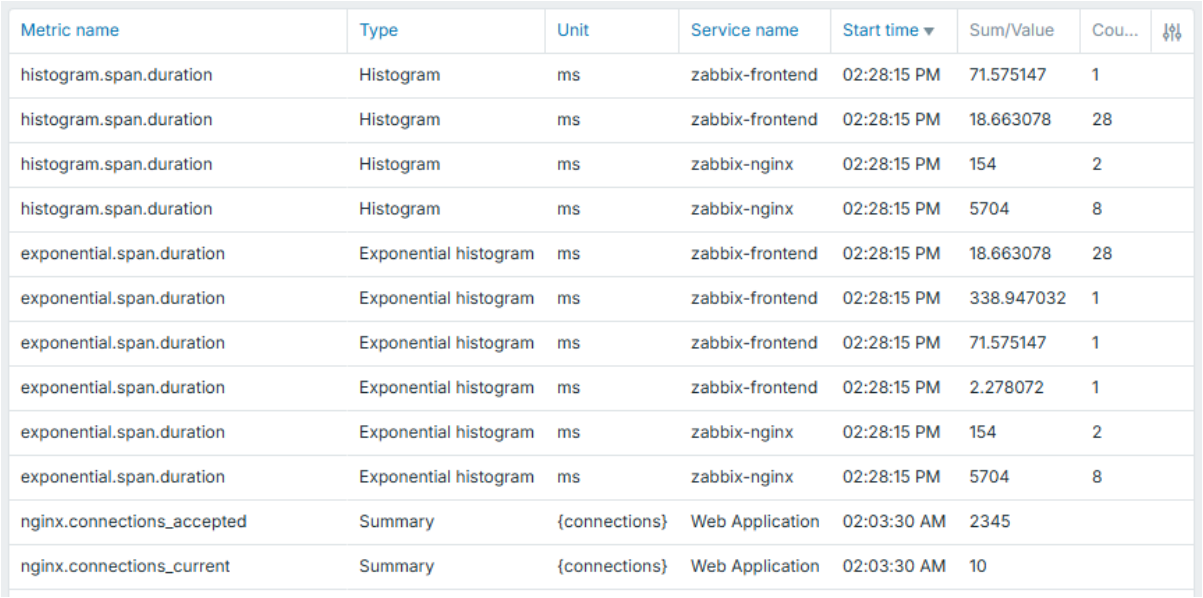


Раздел *APM* > *Метрики* содержит следующие элементы:

- Кнопка режима киоска — отображать только содержимое страницы без меню и заголовка.
- Выбор периода времени — выбрать период времени, за который необходимо отображать точки данных.
- Фильтр — отображать только интересующие вас точки данных.
- Список метрик — просматривать точки данных.
- Сведения о метрике — просматривать сведения о выбранной точке данных.

Список метрик

Каждая строка в списке метрик представляет точку данных.



Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

Точки данных отображаются в настраиваемой таблице. Кнопка *Настроить таблицу* в заголовке таблицы позволяет отображать, скрывать или изменять порядок столбцов, а также восстанавливать макет по умолчанию. Размер каждого столбца также можно изменить, перетащив его правую границу; для сброса размера дважды щёлкните границу. Все изменения применяются немедленно и сохраняются для текущего пользователя.

Список метрик может содержать следующие столбцы:

Note:
Каждый столбец, кроме *Тип*, соответствует столбцу в таблицах метрик (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` и `otel_metrics_exponential_histogram`) источника данных телеметрии; список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии. Большинство столбцов имеют те же названия, что и в источнике данных (например, *Имя метрики* соответствует столбцу `MetricName`). Если названия отличаются, столбец источника данных указан в таблице ниже.

Столбец	Описание
Имя метрики	Метрика, которой принадлежит точка данных (например, `http.server.duration`).
Тип	Тип точки данных: *Gauge*, *Sum*, *Histogram* или *Exponential histogram*. Определяется таблицей метрик, в которой хранится точка данных.
Единица измерения	Единица измерения, если указана, значений метрики (например, `s` или `By`; см. UCUM). Столбец источника данных: `MetricUnit`.
Имя сервиса	Сервис, создавший точку данных.
Имя области	Область инструментирования, создавшая точку данных.
Время начала	Дата и время начала периода измерения точки данных. Столбец источника данных: `StartTimeUnix`.
Сумма/значение	Сумма всех измерений (*Histogram*, *Exponential histogram*) или значение (*Gauge*, *Sum*) точки данных.
Количество	Количество измерений в точке данных типа *Histogram* или *Exponential histogram*.
Время	Дата и время точки данных; для точек данных типов *Sum*, *Histogram* и *Exponential histogram* — конец периода измерения. Столбец источника данных: `TimeUnix`.
Флаги	Флаги точки данных: **0** - флаги не установлены. **1** - для точки данных не записано значение.
Временная характеристика агрегации	Способ агрегации значений точки данных типов *Sum*, *Histogram* или *Exponential histogram* во времени: **Не указано** - не указано. **Delta** - агрегация за период с момента предыдущей точки данных. **Cumulative** - агрегация за период с момента времени начала.
Описание метрики	Описание метрики, если указано.
Атрибуты	Атрибуты точки данных. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.
Атрибуты ресурса	Атрибуты ресурса точки данных. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.

Столбец	Описание
Атрибуты области	Атрибуты области инструментирования. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.

Сведения о метрике

Нажмите точку данных в списке метрик, чтобы открыть панель сведений о метрике, в которой отображается вся информация о точке данных.

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

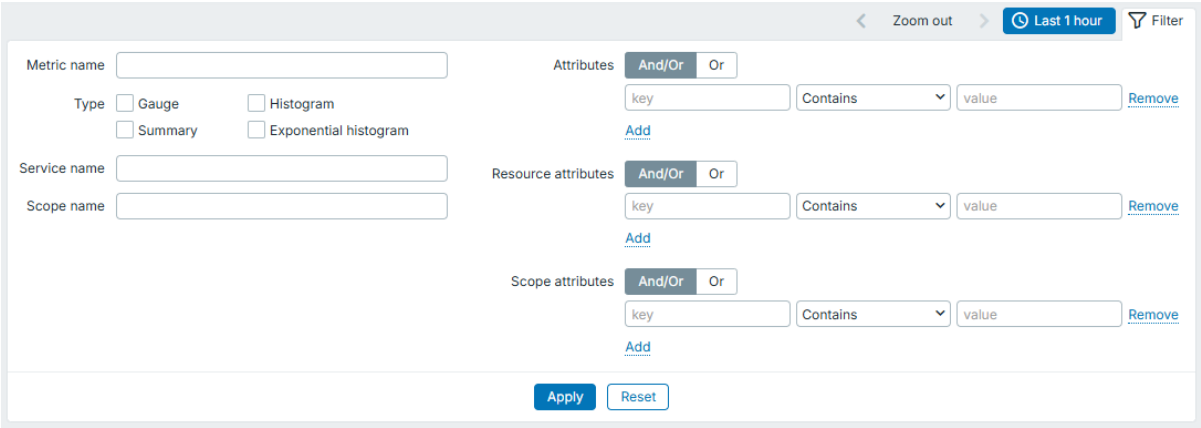
Scope

Панель сведений о метрике может содержать следующие разделы:

Раздел	Описание
Основная информация	Имя метрики, тип, единица измерения, имя сервиса и время начала.
Гистограмма	Интервалы (например, `> 5 – 10`), для каждого из которых указано количество измерений. Отображается только для точек данных типа *Гистограмма*.
Экспоненциальная гистограмма	Интервалы (например, `>= 249.7885 – 249.7887`), для каждого из которых указано количество измерений. Отображается только для точек данных типа *Экспоненциальная гистограмма*.
Информация о метрике	Значения точки данных, зависящие от ее типа, а также описание метрики, время, флаги и URL схемы ресурса.
Атрибуты	Атрибуты точки данных.
Атрибуты ресурса	Атрибуты ресурса.
Атрибуты области	Атрибуты области инструментирования, если они есть.
Область	Имя, версия и URL схемы области инструментирования.

Использование фильтра

С помощью фильтра можно отображать только интересующие вас точки данных.



Чтобы отфильтровать данные по атрибутам, введите ключ атрибута, выберите оператор и введите значение атрибута.

Поддерживаются следующие операторы:

- **Содержит** - *(по умолчанию)* значение атрибута содержит введенную строку.
- **Равно** - значение атрибута равно введенной строке.
- **Не содержит** - значение атрибута не содержит введенную строку.
- **Не равно** - значение атрибута не равно введенной строке.
- **Существует** - атрибут с введенным ключом существует.
- **Не существует** - атрибут с введенным ключом не существует.

Если вы добавили несколько условий, выберите способ их оценки:

- **И/Или** - *(по умолчанию)* должны выполняться все условия; условия с одинаковым ключом атрибута объединяются с помощью *Или*.
- **Или** - должно выполняться хотя бы одно условие.

Журналы

Обзор

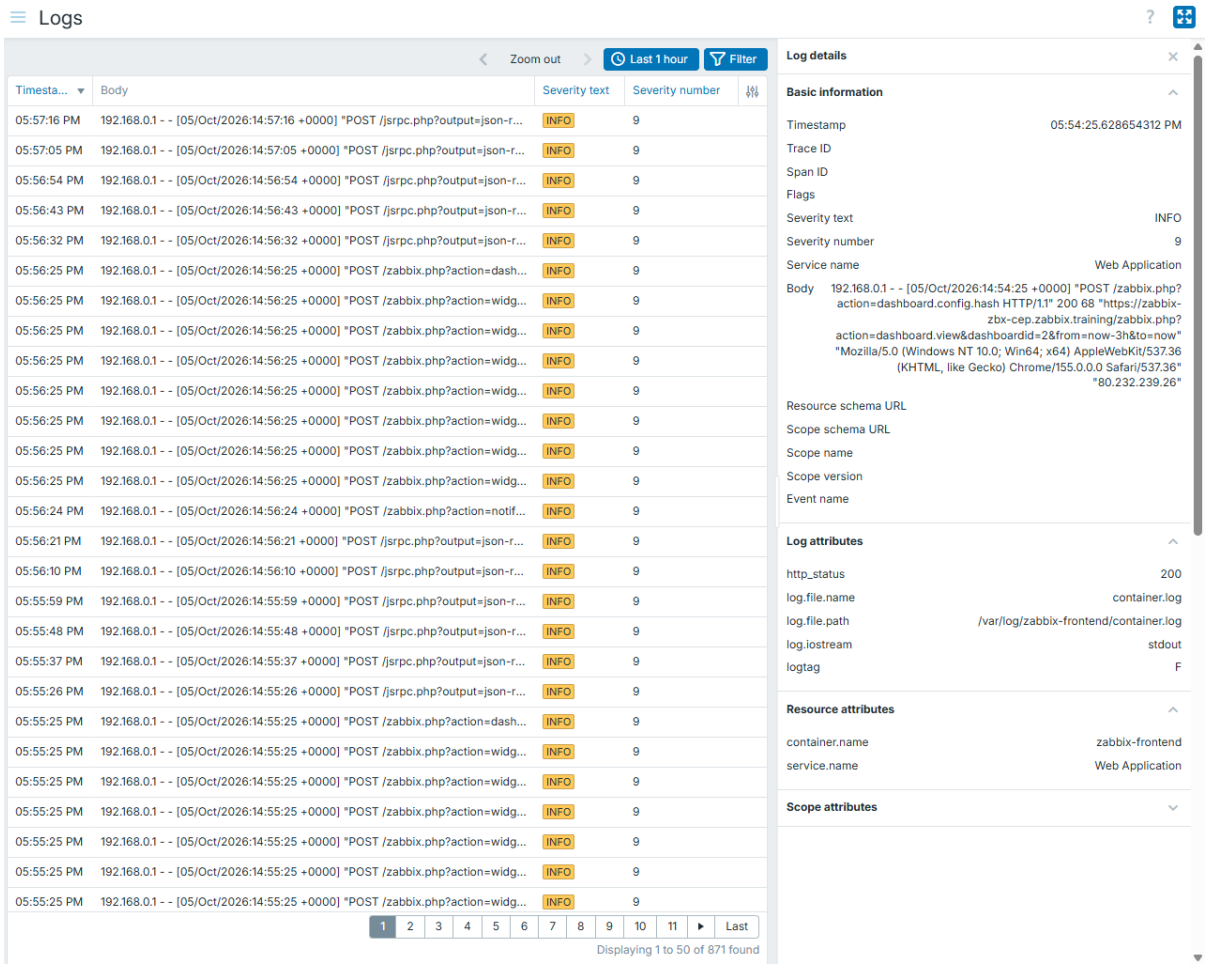
В разделе *APM* > *Журналы* отображаются записи журналов, собранные из ваших приложений и служб.

Запись журнала — это текстовая запись о событии (например, о неудачном платеже или превышении времени ожидания базы данных). Каждая запись журнала содержит уровень серьезности и сообщение (тело). Если запись журнала была создана во время операции, являющейся частью трассировки, она также содержит идентификаторы этой трассировки и спана.

Каждая запись журнала создается **службой** (например, `zabbix-frontend`; имя определяется атрибутом ресурса Open-Telemetry `service.name`) и областью инструментирования — частью кода, создавшей эту запись (например, библиотекой ведения журналов).

Для предоставления контекста о каждом событии записи журналов также содержат **атрибуты** «ключ-значение»: атрибуты журнала описывают само событие (например, `log.file.path: /var/log/zabbix/zabbix_frontend.log`), а атрибуты ресурса описывают службу и среду, в которой она работает (например, `host.name: eabf8c2a5ddc`).

Дополнительную информацию о журналах см. в документации OpenTelemetry.

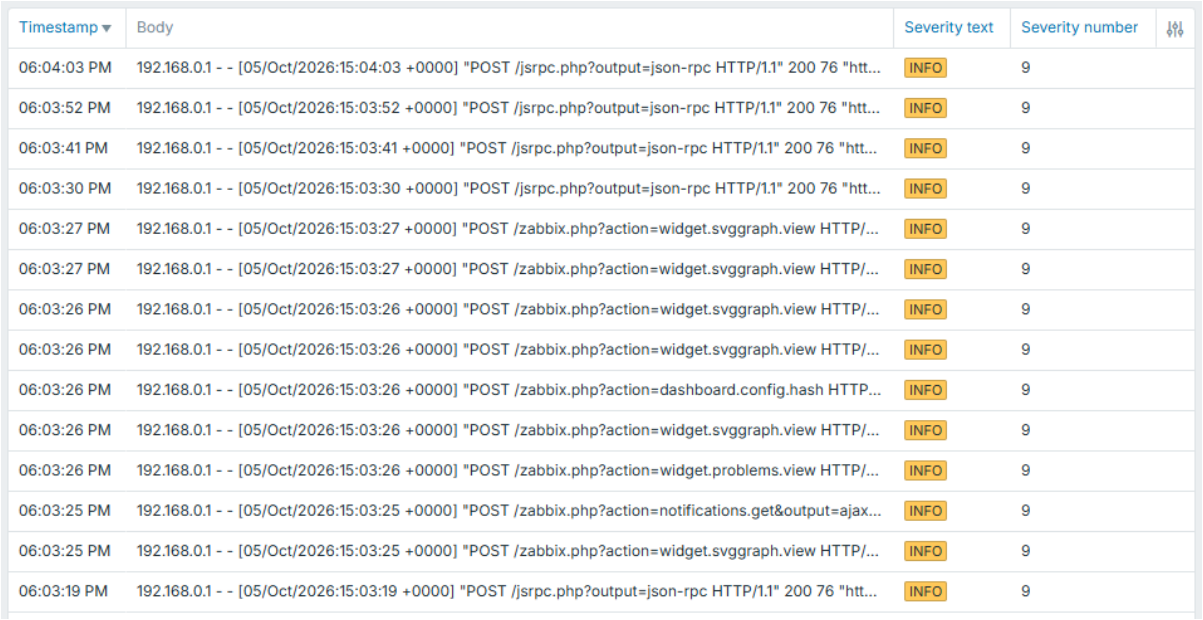


Раздел *APM* > *Журналы* содержит следующие элементы:

- кнопка режима киоска — отображать только содержимое страницы без меню и заголовка;
- выбор периода времени — выбрать период времени, за который необходимо отображать записи журналов;
- фильтр — отображать только интересующие вас записи журналов;
- список журналов — просматривать записи журналов;
- сведения о журнале — просматривать сведения о выбранной записи журнала.

Список журналов

Каждая строка в списке журналов представляет запись журнала.



Timestamp ▾	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

Записи журналов отображаются в настраиваемой таблице. Кнопка *Настроить таблицу* в заголовке таблицы позволяет отображать, скрывать или изменять порядок столбцов, а также восстанавливать макет по умолчанию. Размер каждого столбца также можно изменить, перетащив его правую границу; для восстановления размера по умолчанию дважды щелкните границу. Все изменения применяются немедленно и сохраняются для текущего пользователя.

Список журналов может содержать следующие столбцы:

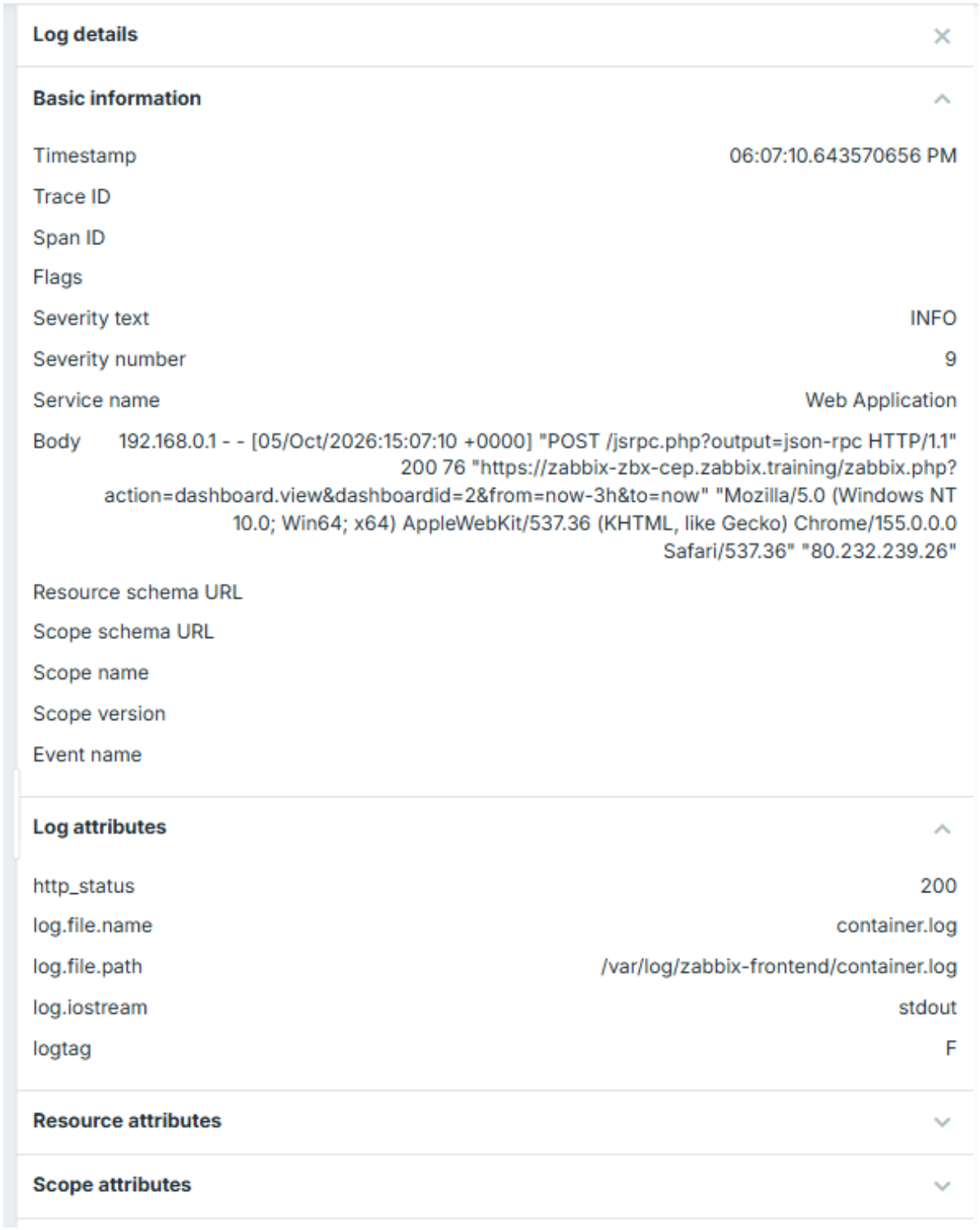
Note:
Каждый столбец соответствует столбцу в таблице `otel_logs` источника данных телеметрии; список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии. Большинство столбцов имеют те же названия, что и в источнике данных (например, *Текст уровня серьезности* соответствует столбцу `SeverityText`). Если названия отличаются, столбец источника данных указан в таблице ниже.

Столбец	Описание
Метка времени	Дата и время возникновения события.
Тело	Сообщение записи журнала.
Текст уровня серьезности	Уровень серьезности, указанный приложением (например, `INFO`, `WARN` или `ERROR`). Цвет метки зависит от уровня серьезности.
Числовое значение уровня серьезности	Уровень серьезности в виде числа: **0** - НЕ УКАЗАН. **1-4** - ТРАССИРОВКА. **5-8** - ОТЛАДКА. **9-12** - ИНФОРМАЦИЯ. **13-16** - ПРЕДУПРЕЖДЕНИЕ. **17-20** - ОШИБКА. **21-24** - КРИТИЧЕСКАЯ ОШИБКА.
ID трассировки	ID трассировки, во время которой была создана запись журнала, если он имеется.
ID спана	ID спана, во время которого была создана запись журнала, если он имеется.
Флаги	Флаги трассировки записи журнала: **0** - трассировка не отбирается. **1** - трассировка отбирается (записывается). Столбец источника данных: `TraceFlags`.
Имя сервиса	Сервис, создавший запись журнала.
Имя области	Область инструментирования, создавшая запись журнала.
Версия области	Версия области инструментирования.
Имя события	Имя типа события, представленного записью журнала, если оно имеется.
Атрибуты журнала	Атрибуты журнала записи журнала. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.
Атрибуты ресурса	Атрибуты ресурса записи журнала. Нажмите кнопку с тремя точками ••• , чтобы отобразить все атрибуты.

Столбец	Описание
Атрибуты области	Атрибуты области инструментирования. Нажмите кнопку с тремя точками ⋯, чтобы отобразить все атрибуты.

Сведения о журнале

Нажмите запись журнала в списке журналов, чтобы открыть панель *Сведения о журнале*, в которой отображается вся информация о записи журнала.



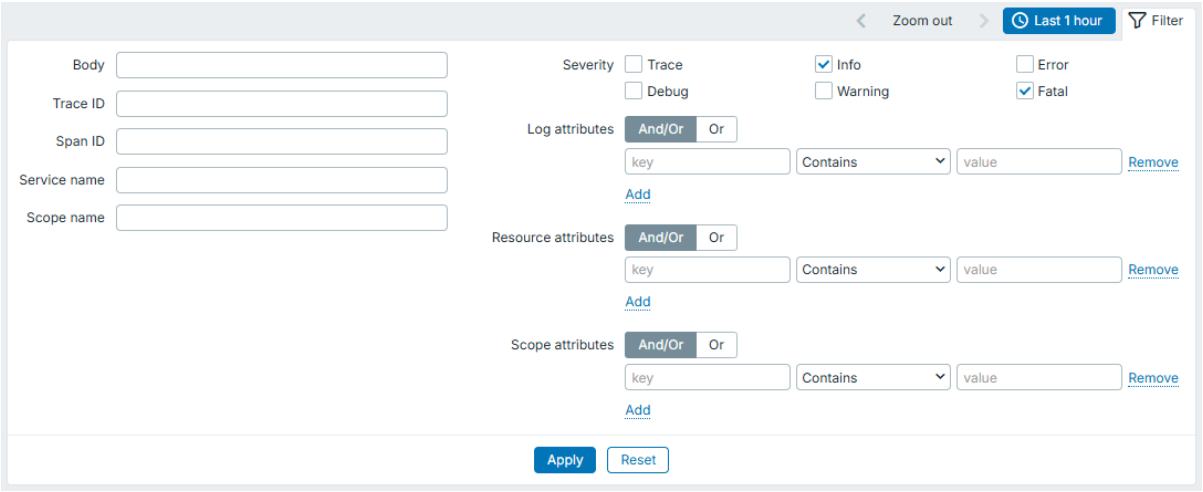
Панель *Сведения о журнале* может содержать следующие разделы:

Раздел	Описание
Основная информация	Идентификатор трассировки, идентификатор span и флаги трассировки записи журнала.
Атрибуты журнала	Атрибуты журнала записи журнала.

Раздел	Описание
Атрибуты ресурса	URL схемы и атрибуты ресурса.
Атрибуты области	URL схемы, имя, версия и атрибуты области инструментирования.

Использование фильтра

С помощью фильтра можно отображать только интересующие вас записи журнала.



Чтобы отфильтровать записи по атрибутам, введите ключ атрибута, выберите оператор и введите значение атрибута.

Поддерживаются следующие операторы:

- **Содержит** - *(по умолчанию)* значение атрибута содержит введенную строку.
- **Равно** - значение атрибута равно введенной строке.
- **Не содержит** - значение атрибута не содержит введенную строку.
- **Не равно** - значение атрибута не равно введенной строке.
- **Существует** - атрибут с введенным ключом существует.
- **Не существует** - атрибут с введенным ключом не существует.

Если вы добавили несколько условий, выберите способ их оценки:

- **И/Или** - *(по умолчанию)* должны выполняться все условия; условия с одинаковым ключом атрибута объединяются оператором *Или*.
- **Или** - должно выполняться хотя бы одно условие.

Службы

Обзор

Меню ”Услуги” предназначено для функций мониторинга услуг Zabbix.

Службы

Обзор

В этом разделе можно просмотреть общую информацию о состоянии всех служб, настроенных в Zabbix с учетом вашей инфраструктуры.

Служба может представлять собой иерархию, состоящую из нескольких уровней других служб, называемых дочерними службами. Они влияют на общее состояние службы (см. также обзор функциональности мониторинга служб).

Основные категории состояния службы: *OK* или *Проблема*. Состояние *Проблема* отображается с использованием соответствующего названия и цвета уровня серьезности проблемы.

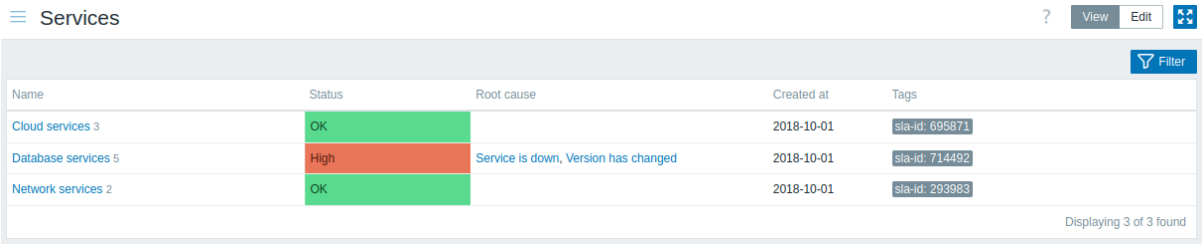
В режиме просмотра можно отслеживать службы, их состояние и другие сведения. Кроме того, в этом разделе можно настроить иерархию служб (добавлять и редактировать службы и дочерние службы), переключившись в режим редактирования.

Чтобы переключиться из режима просмотра в режим редактирования (и обратно), нажмите соответствующую кнопку в правом верхнем углу:

- View - просмотр служб
- Edit - добавление и редактирование служб и дочерних служб

Обратите внимание, что доступ к редактированию зависит от настроек роли пользователя.

Просмотр сервисов



Отображается список существующих сервисов.

Отображаемые данные:

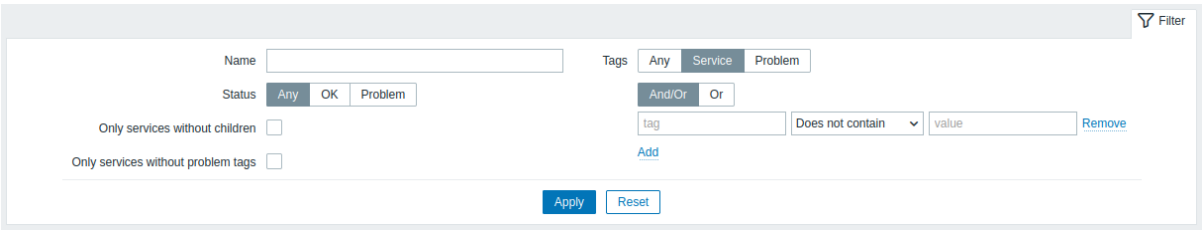
Parameter	Description
Name	Имя сервиса. Имя сервиса является ссылкой на сведения о сервисе. Число после имени показывает, сколько дочерних сервисов у сервиса.
Status	Состояние сервиса: **OK** - проблем нет **<problem color and severity>** - указывает на проблему и ее серьезность. При наличии нескольких проблем отображаются цвет и серьезность наиболее критичной проблемы.
Root cause	Перечислены основные проблемы, которые напрямую или косвенно влияют на состояние сервиса. Те же проблемы перечислены в соответствии с {SERVICE.ROOTCAUSE} макросом. Щелкните имя проблемы, чтобы увидеть дополнительные сведения о ней в разделе *Monitoring* > *Problems*. Проблемы, которые не влияют на состояние сервиса, в список не включаются.
Created at	Отображается время, когда сервис был создан.
Tags	Отображаются теги сервиса. Теги используются для идентификации сервиса в действиях сервиса и SLA.

Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас сервисы.



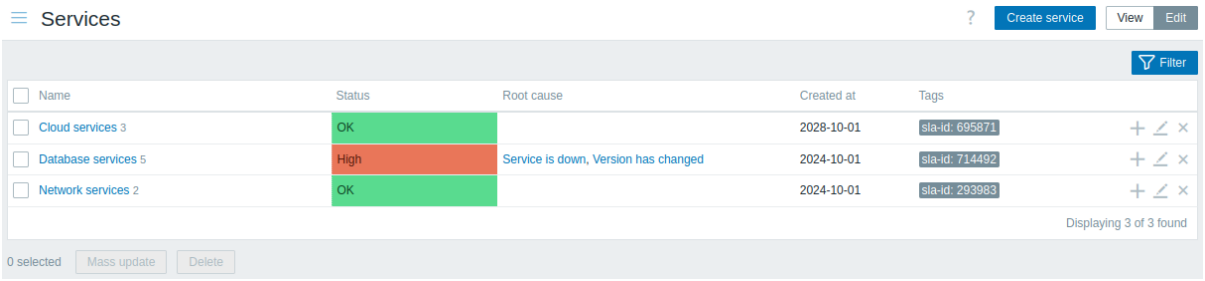
Parameter	Description
Name	Фильтр по имени сервиса.
Status	Фильтр по состоянию сервиса.
Only services without children	Отметьте флажок, чтобы отображать только сервисы без дочерних сервисов. Этот параметр доступен только в режиме редактирования сервиса.
Only services without problem tags	Отметьте флажок, чтобы отображать только сервисы без тегов проблем. Этот параметр доступен только в режиме редактирования сервиса.

Parameter	Description
Tags	Фильтр по имени и значению тега сервиса или по имени и значению тега проблемы сервиса (в режиме редактирования сервиса). Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно выполнения одного условия.

Редактирование услуг

Нажмите кнопку *Edit*, чтобы перейти в режим редактирования. В режиме редактирования список дополняется флажками перед записями, а также следующими дополнительными параметрами:

-  - добавить дочернюю услугу к этой услуге
-  - редактировать эту услугу
-  - удалить эту услугу

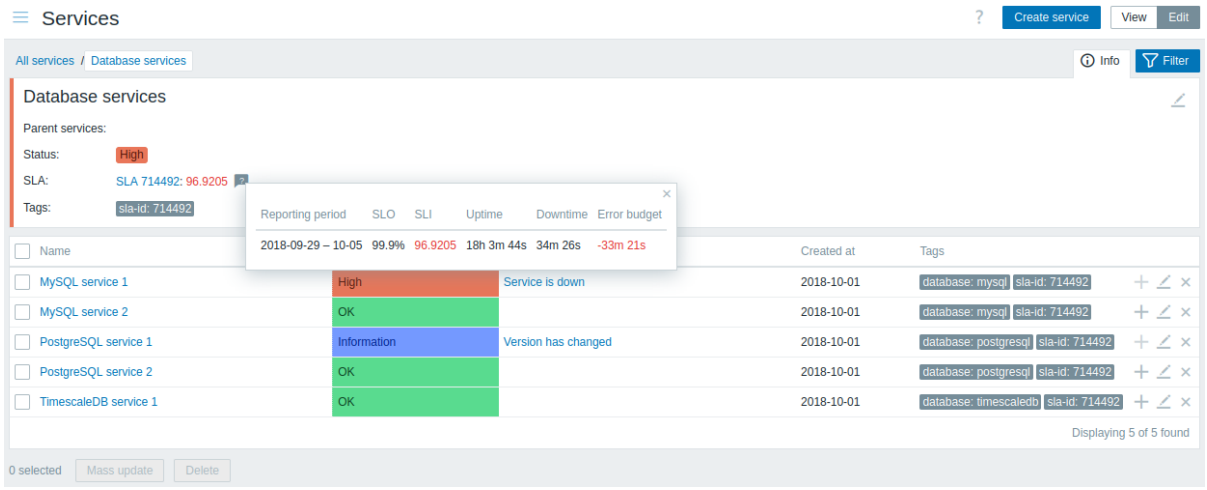


Чтобы настроить новую услугу, нажмите кнопку *Create service* в правом верхнем углу.

Сведения о сервисе

Чтобы открыть сведения о сервисе, нажмите на имя сервиса. Чтобы вернуться к списку всех сервисов, нажмите *Все сервисы*.

Сведения о сервисе включают информационный блок и список дочерних сервисов.



Чтобы открыть информационный блок, нажмите вкладку *Info*. Информационный блок содержит следующие поля:

- Имена родительских сервисов (если есть)
- Текущий статус этого сервиса
- Текущие SLA этого сервиса в формате `SLA name:service level indicator`. 'SLA name' также является ссылкой на отчет SLA для этого сервиса. Если навести указатель мыши на информационный блок рядом с индикатором уровня сервиса (SLI), отобразится всплывающий список с подробностями SLI. Индикатор уровня сервиса показывает текущий уровень сервиса в процентах.
- Теги сервиса

Информационный блок также содержит ссылку на конфигурацию сервиса.

Чтобы использовать фильтр для дочерних сервисов, нажмите вкладку *Filter*.

В режиме редактирования список дочерних сервисов дополняется дополнительными параметрами редактирования:

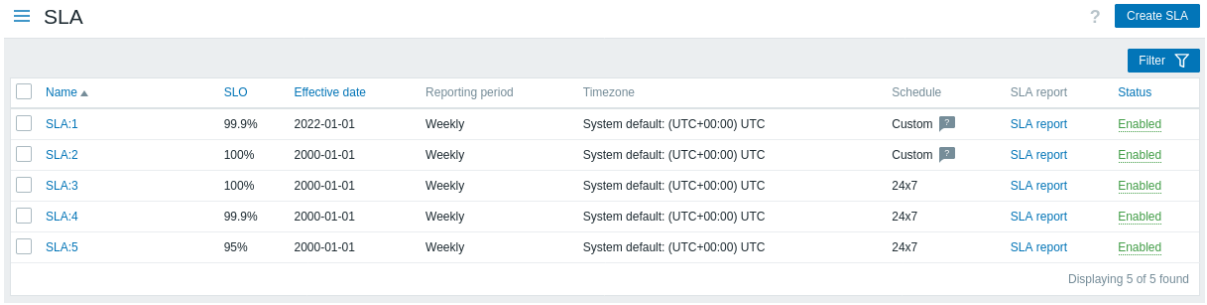
-  - добавить дочерний сервис к этому сервису
-  - редактировать этот сервис
-  - удалить этот сервис

SLA

Обзор

В этом разделе можно просматривать и настраивать SLA.

SLA



Отображается список настроенных SLA. *Примечание*: будут отображаться только те SLA, которые относятся к сервисам, доступным пользователю (только для чтения, если для роли пользователя не включено *Управление SLA*).

Отображаемые данные:

Parameter	Description
Name	Отображается имя SLA. Имя является ссылкой на настройку SLA.
SLO	Отображается целевой уровень обслуживания (SLO).
Effective date	Отображается дата начала расчета SLA.

Parameter	Description
Reporting period	Отображается период, используемый в отчете SLA: *ежедневно*, *еженедельно*, *ежемесячно*, *ежеквартально* или *ежегодно*.
Time zone	Отображается часовой пояс SLA.
Schedule	Отображается расписание SLA: 24x7 или пользовательское.
SLA report	Щелкните ссылку, чтобы просмотреть отчет SLA для этого SLA.
Status	Отображается статус SLA: включено или отключено.

Отчет SLA

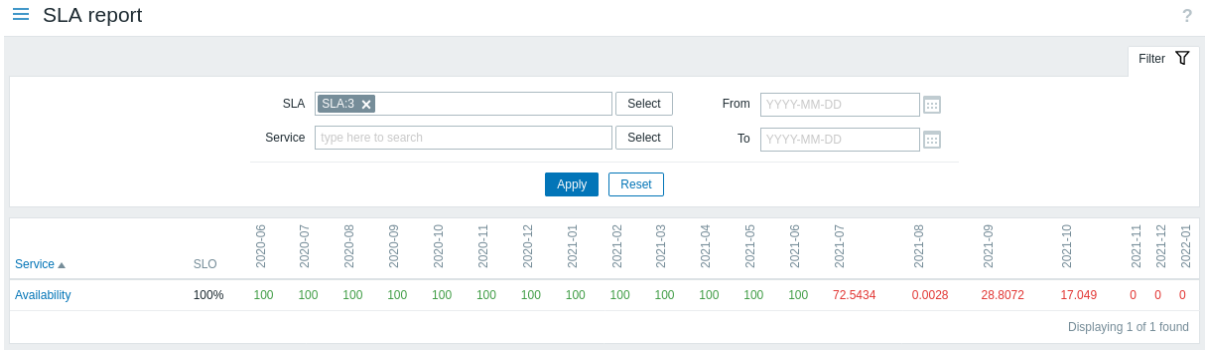
Обзор

В этом разделе можно просматривать отчеты SLA на основе критериев, выбранных в фильтре.

Отчеты SLA также можно отображать в виде виджета панели мониторинга.

Отчет

Фильтр позволяет выбрать отчет по имени SLA, а также по имени сервиса. Кроме того, можно ограничить отображаемый период.

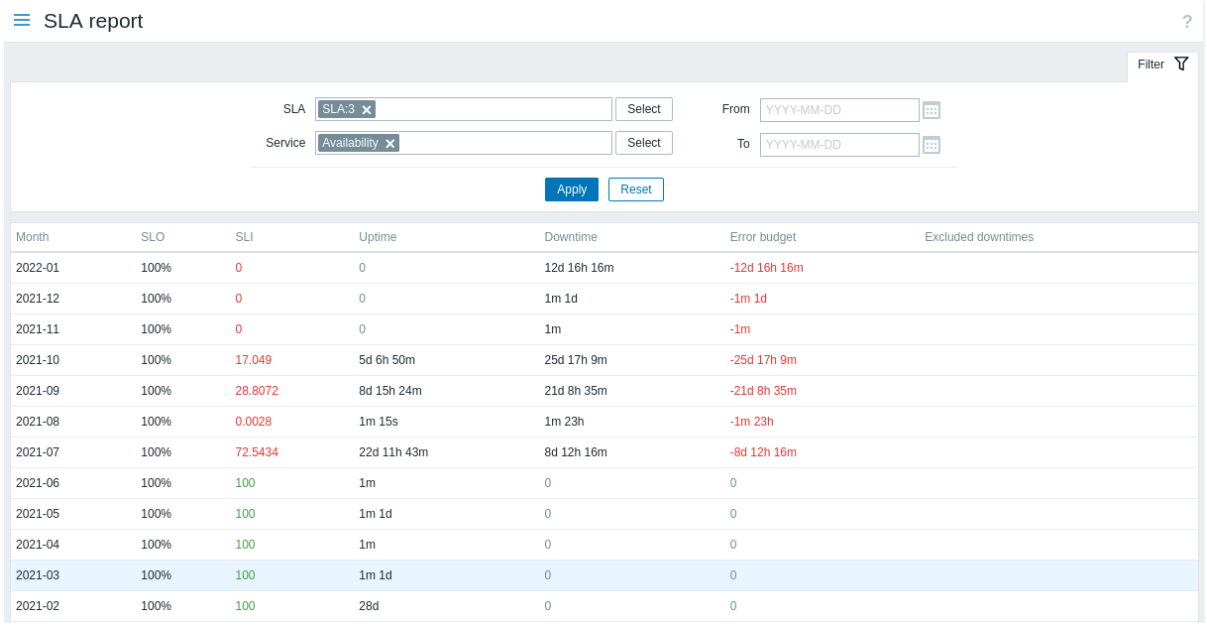


В каждом столбце (периоде) отображается SLI за соответствующий период. Значения SLI, нарушающие заданное SLO, выделяются красным цветом.

В отчете отображается 20 периодов. Можно отобразить не более 100 периодов, если указаны даты *From* и *To*.

Детали отчета

Если вы нажмете на имя сервиса в отчете, вы сможете открыть другой отчет, который отображает более подробное представление.



Обратите внимание, что отрицательная длительность проблемы не влияет на расчет SLA или формирование отчетов.

Инвентаризация

Обзор

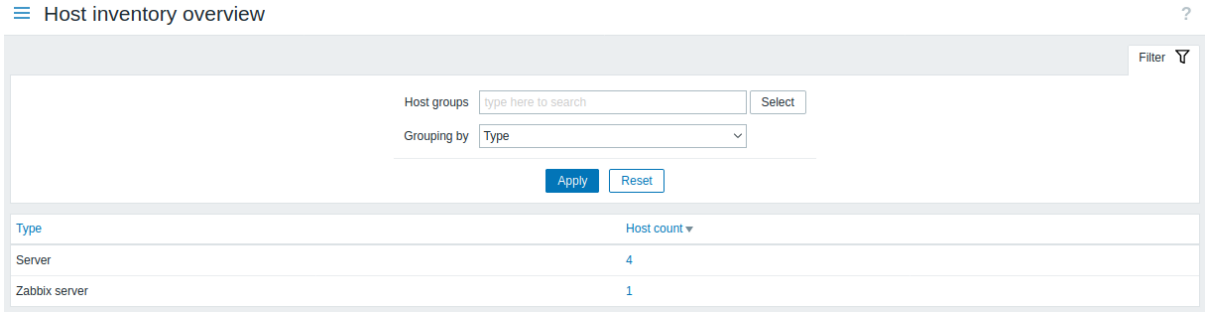
Меню «Инвентаризация» содержит разделы, предоставляющие обзор данных инвентаризации узлов сети по выбранному параметру, а также возможность просматривать подробные сведения об инвентаризации узлов сети.

Обзор

Обзор

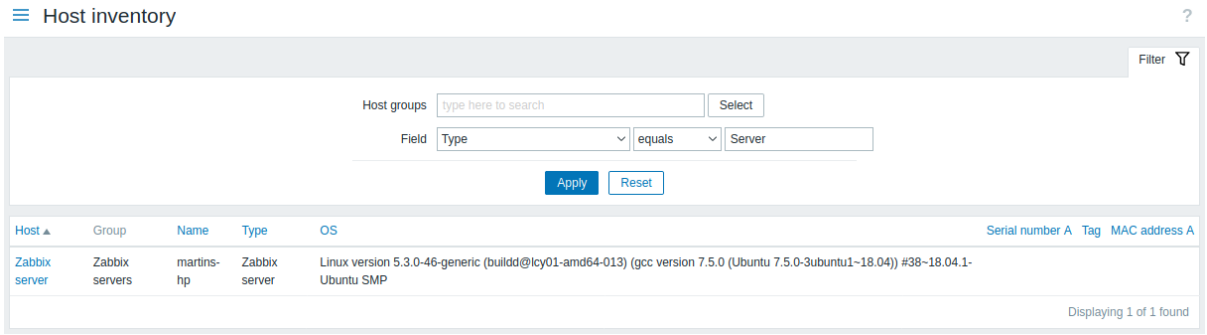
Раздел *Inventory* > *Overview* предоставляет способы получения общего представления о данных инвентаризации узла сети.

Чтобы отобразить обзор, выберите группы узлов сети (или не выбирайте ни одной) и поле инвентаризации, по которому нужно отображать данные. Будет показано количество узлов сети, соответствующих каждому значению выбранного поля.



Полнота обзора зависит от того, насколько полно сведения инвентаризации заполнены для узлов сети.

Числа в столбце *Host count* являются ссылками; они приводят к фильтрации этих узлов сети в таблице *Host Inventories*.

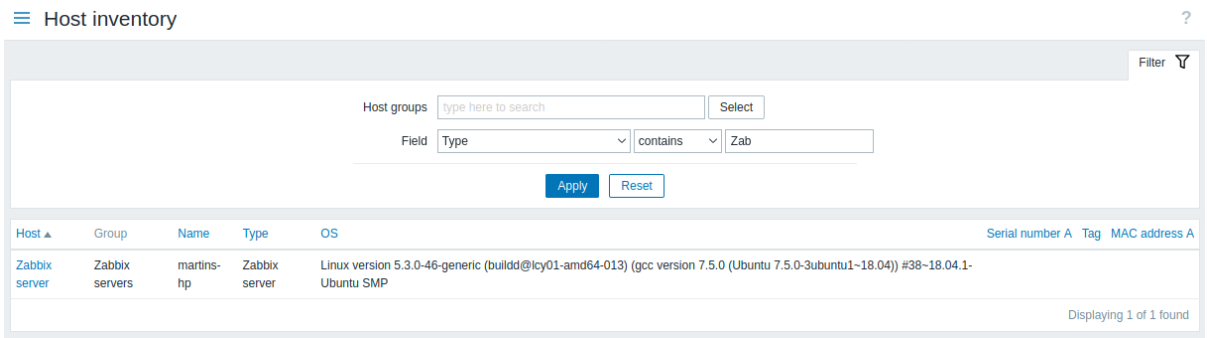


Узлы сети

Обзор

В разделе *Inventory* > *Hosts* данные инвентаризации узлов сети отображаются.

Вы можете фильтровать узлы сети по группам узлов сети и по любому полю инвентаризации, чтобы отображать только интересующие вас узлы сети.

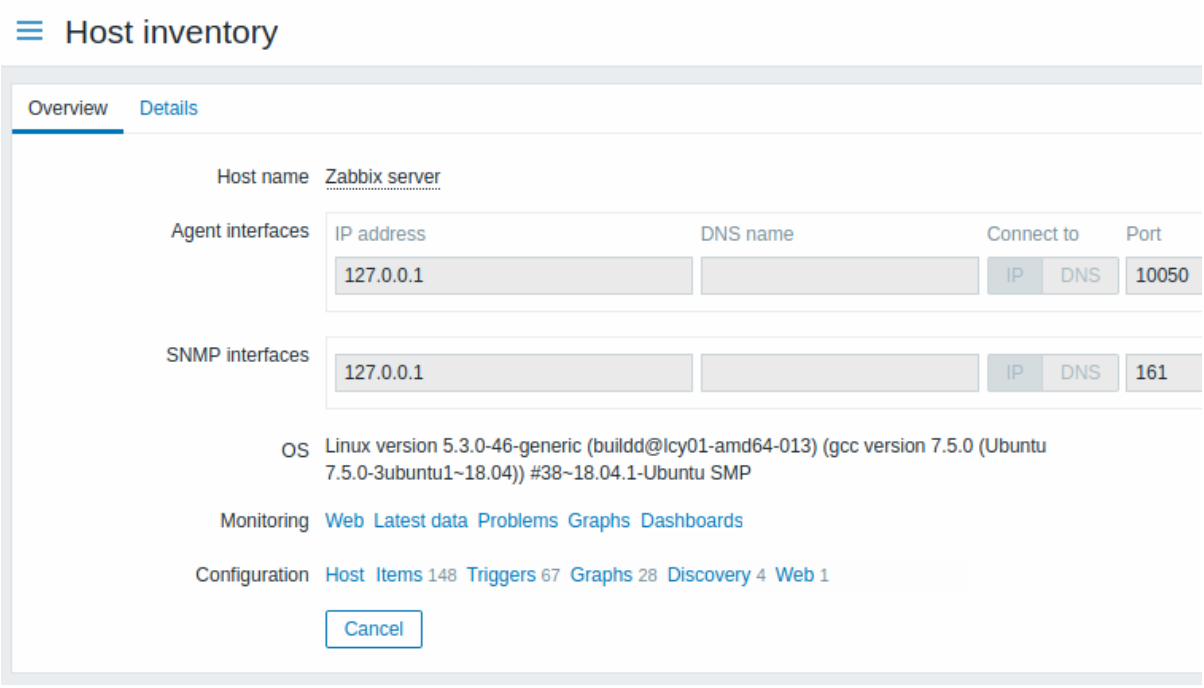


Чтобы отобразить всю инвентаризацию узлов сети, нажмите кнопку *Reset*.

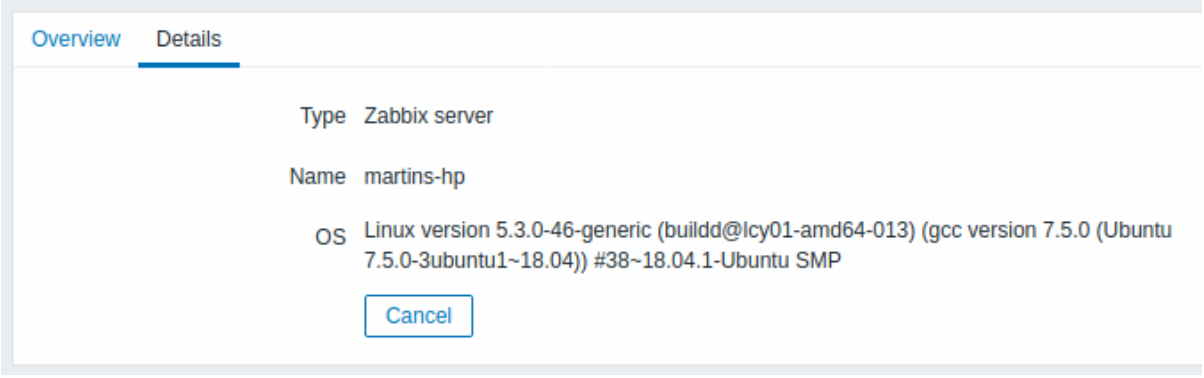
Хотя в таблице отображаются только некоторые ключевые поля инвентаризации, вы также можете просмотреть всю доступную информацию инвентаризации для этого узла сети. Для этого нажмите имя узла сети в первом столбце.

Сведения об инвентаризации

Вкладка **Обзор** содержит общую информацию об узле сети, последние данные мониторинга и параметры конфигурации узла сети:



Вкладка **Подробности** содержит все доступные сведения об инвентаризации для узла сети:



Полнота данных инвентаризации зависит от того, насколько полно ведется информация об инвентаризации для узла сети. Если информация не ведется, вкладка *Подробности* отключена.

Отчеты

Обзор

В меню «Отчеты» есть несколько разделов, содержащих множество предопределенных и настраиваемых пользователем отчетов, ориентированных на отображение таких параметров, как системная информация, триггеры и собранные данные.

Системная информация

Обзор

В разделе *Отчёты* > *Сведения о системе* отображается сводка основных данных о сервере Zabbix и системе. Системные данные собираются с помощью внутренних элементов данных.

Отчёт со сведениями о системе виден только пользователям Zabbix с ролями *Суперадминистратор* и *Администратор*.

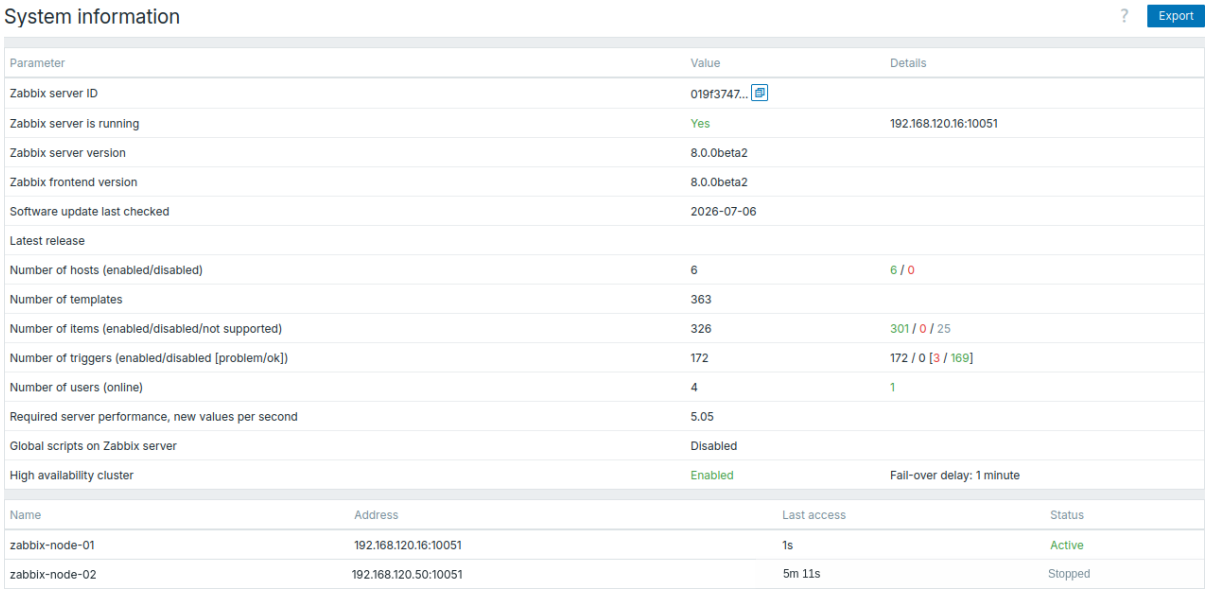
Обратите внимание, что в конфигурации высокой доступности можно перенаправить источник сведений о системе (экземпляр сервера). Для этого отредактируйте файл конфигурации веб-интерфейса — раскомментируйте и задайте

$ZBX_SERVER или одновременно $ZBX_SERVER и $ZBX_SERVER_PORT, указав сервер, отличный от отображаемого активного сервера. Обратите внимание: если задан только $ZBX_SERVER, для $ZBX_SERVER_PORT будет использоваться значение по умолчанию (10051).

Если конфигурация высокой доступности включена, под статистикой системы отображается отдельный блок со сведениями об узлах высокой доступности.

Сведения о системе также доступны в виде виджета панели мониторинга.

Системная статистика



Отображаемые данные:

Параметр	Значение	Подробности
Идентификатор сервера Zabbix	Уникальный идентификатор сервера Zabbix (UUIDv7).	Идентификатор сервера генерируется при первом запуске сервера. Отображаются первые 8 символов, за которыми следует многоточие. При наведении курсора на значение идентификатора сервера или нажатии на него появляется всплывающая подсказка с полным значением. Нажатие на значок рядом со значением копирует идентификатор в буфер обмена. Если включена высокая доступность, каждый узел имеет собственный идентификатор узла для адресации из командной строки, но все узлы объединены под одним идентификатором сервера, сгенерированным узлом, который был запущен первым (или ранее являлся автономным сервером), независимо от того, какой узел активен в данный момент.
Сервер Zabbix запущен	Состояние сервера Zabbix: **Да** - сервер запущен **Нет** - сервер не запущен *Примечание:* Для отображения остальной информации веб-интерфейсу необходимо, чтобы сервер был запущен и на сервере был запущен как минимум один процесс траппера (параметр StartTrappers в файле zabbix_server.conf > 0).	Расположение и порт сервера Zabbix.

Параметр	Значение	Подробности
Версия сервера Zabbix	Отображается текущий номер версии сервера. *Примечание:* Отображается только при запущенном сервере Zabbix.	Отображается состояние версии сервера: **Актуальная** - используется последняя версия; **Доступно новое обновление** - доступна более новая версия; **Устаревшая** - период полной поддержки этой версии завершен. Эта информация доступна только если проверка обновлений программного обеспечения включена в конфигурации сервера Zabbix. Ничего не отображается, если последняя проверка обновлений программного обеспечения выполнялась более недели назад или отсутствуют данные о текущей версии.
Версия веб-интерфейса Zabbix	Отображается номер версии веб-интерфейса Zabbix.	Отображается состояние версии веб-интерфейса Zabbix: **Актуальная** - используется последняя версия; **Доступно новое обновление** - доступна более новая версия; **Устаревшая** - период полной поддержки этой версии завершен. Эта информация доступна только если проверка обновлений программного обеспечения включена в конфигурации сервера Zabbix. Ничего не отображается, если последняя проверка обновлений программного обеспечения выполнялась более недели назад или отсутствуют данные о текущей версии.
Последняя проверка обновлений программного обеспечения	Отображается дата последней проверки обновлений программного обеспечения Zabbix. Эта информация доступна только если проверка обновлений программного обеспечения включена в конфигурации сервера.	
Последний выпуск	Отображается номер более нового выпуска (если доступен) для текущей версии Zabbix. Эта информация доступна только если проверка обновлений программного обеспечения включена в конфигурации сервера. Ничего не отображается, если последняя проверка обновлений программного обеспечения выполнялась более недели назад или отсутствуют данные о текущей версии.	Отображается ссылка на примечания к выпуску последнего доступного выпуска Zabbix.
Количество узлов сети	Отображается общее количество настроенных узлов сети.	Количество отслеживаемых и не отслеживаемых узлов сети.
Количество шаблонов	Отображается общее количество шаблонов.	
Количество элементов данных	Отображается общее количество элементов данных.	Количество отслеживаемых, отключенных и неподдерживаемых элементов данных на уровне узлов сети. Элементы данных на отключенных узлах сети учитываются как отключенные.
Количество триггеров	Отображается общее количество триггеров.	Количество включенных и отключенных триггеров на уровне узлов сети; распределение включенных триггеров по состояниям «Проблема» и «OK». Триггеры, перечисленные в состоянии «OK», включают триггеры со статусом «Неизвестно». Триггеры, зависящие от отключенных элементов данных или назначенные отключенным узлам сети, учитываются как отключенные.

Параметр	Значение	Подробности
Количество пользователей	Отображается общее количество настроенных пользователей.	Количество пользователей в сети.
Требуемая производительность сервера, новых значений в секунду	Отображается ожидаемое количество новых значений, обрабатываемых сервером Zabbix в секунду.	*Требуемая производительность сервера* является оценкой и может использоваться в качестве ориентира. Для получения точных значений количества обработанных значений используйте `zabbix[wcache,values,all]` внутренний элемент данных. В расчет включаются включенные элементы данных отслеживаемых узлов сети. Элементы данных журнала учитываются как одно значение за каждый интервал обновления элемента данных. Учитываются значения с обычным интервалом; значения с гибким интервалом и интервалом планирования не учитываются. Расчет не корректируется в течение периода обслуживания «нет данных». Элементы данных-трапперы не учитываются.
Глобальные скрипты на сервере Zabbix	В этом поле отображается **Отключено**, если глобальные скрипты отключены на сервере Zabbix путем установки EnableGlobalScripts=0 в конфигурации сервера.	
Кластер высокой доступности	Состояние кластера высокой доступности сервера Zabbix: **Отключен** - автономный сервер **Включен** - существует как минимум один узел высокой доступности	Если кластер включен, отображается задержка переключения при отказе.

В разделе *Системная информация* также отображается сообщение об ошибке при следующих условиях:

- Используемая база данных не имеет требуемой кодировки или сортировки (UTF-8).
- Версия базы данных ниже или выше поддерживаемого диапазона (доступно только пользователям с типом роли *Суперадминистратор*).
- Очистка данных для TimescaleDB настроена неправильно (таблицы истории или трендов содержат сжатые фрагменты, но параметры *Переопределить период хранения истории элементов данных* или *Переопределить период хранения трендов элементов данных* отключены).

Кнопки

Кнопка в правом верхнем углу страницы предлагает следующую опцию:

Export	Экспортировать системную информацию в JSON-файл. Файл сохраняется как `Zabbix-system-information-{time}.json`, где `{time}` — это временная метка Unix, указывающая, когда данные были подготовлены на сервере. Если включена высокая доступность, также экспортируются сведения об узлах высокой доступности.

Узлы высокой доступности

Если кластер высокой доступности включен, отображается дополнительный блок данных, показывающий статус каждого узла высокой доступности.



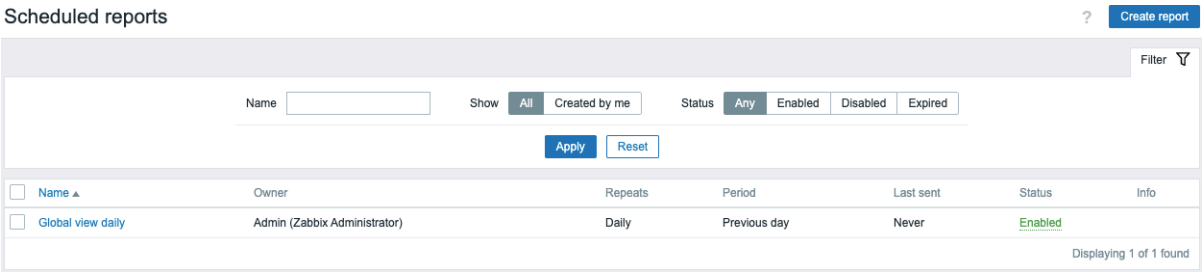
Отображаемые данные:

Column	Description
Name	Имя узла, как определено в конфигурации сервера.
Address	IP-адрес и порт узла.
Last access	Время последнего доступа к узлу. При наведении курсора на ячейку отображается метка времени последнего доступа в полном формате.
Status	Статус узла. Строки таблицы отсортированы по этим статусам в следующем порядке приоритета: **Active** - узел работает и функционирует **Unavailable** - узел не был обнаружен дольше, чем задержка переключения при отказе (возможно, стоит выяснить причину) **Stopped** - узел был остановлен или не смог запуститься (возможно, стоит запустить его или удалить) **Standby** - узел работает и ожидает

Запланированные отчеты

Обзор

В разделе *Отчёты* > *Запланированные отчёты* пользователи с достаточными правами могут настроить регулярное создание PDF-версий панелей, которые будут отправляться по электронной почте указанным получателям.



На начальном экране отображается информация о запланированных отчётах, которую можно отфильтровать для удобной навигации — см. раздел Использование фильтра ниже.

Отображаемые данные:

Столбец	Описание
Имя	Имя отчёта. При нажатии открывается форма настройки отчёта.
Владелец	Пользователь, создавший отчёт.
Повторение	Частота создания отчёта (ежедневно/еженедельно/ежемесячно/ежегодно).
Период	Период времени, за который подготавливается отчёт (предыдущий день, неделю, месяц или год).
Последняя отправка	Дата и время отправки последнего отчёта.
Статус	Текущий статус отчёта (включён/отключён/истёк). Пользователи с достаточными правами могут изменить статус, нажав на него: с «Включён» на «Отключён» (и обратно); с «Истёк» на «Отключён» (и обратно). Для пользователей с недостаточными правами статус недоступен для нажатия.

Столбец	Описание
Информация	Отображает информационные значки: Красный значок указывает, что создание отчёта завершилось ошибкой; при наведении на него отображается всплывающая подсказка с информацией об ошибке. Жёлтый значок указывает, что отчёт был создан, но его отправка некоторым (или всем) получателям завершилась ошибкой, либо что срок действия отчёта истёк; при наведении на него отображается всплывающая подсказка с дополнительной информацией.

Использование фильтра

Вы можете использовать фильтр, чтобы сузить список отчетов. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Доступны следующие параметры фильтрации:

- *Имя* - допускается частичное совпадение имени
- *Показать* - созданные текущим пользователем или все отчеты
- *Статус* - выберите один из вариантов: ”Any” (показать все отчеты), ”Enabled”, ”Disabled” или ”Expired”

Фильтр расположен под названием раздела *Scheduled reports*. Его можно открыть и свернуть, нажав вкладку *Filter* в правом верхнем углу.

Массовое обновление

Иногда может потребоваться удалить или изменить статус сразу нескольких отчетов. Вместо того чтобы открывать каждый отчет по отдельности для редактирования, для этого можно использовать функцию массового обновления.

Чтобы массово обновить несколько отчетов, выполните следующие действия:

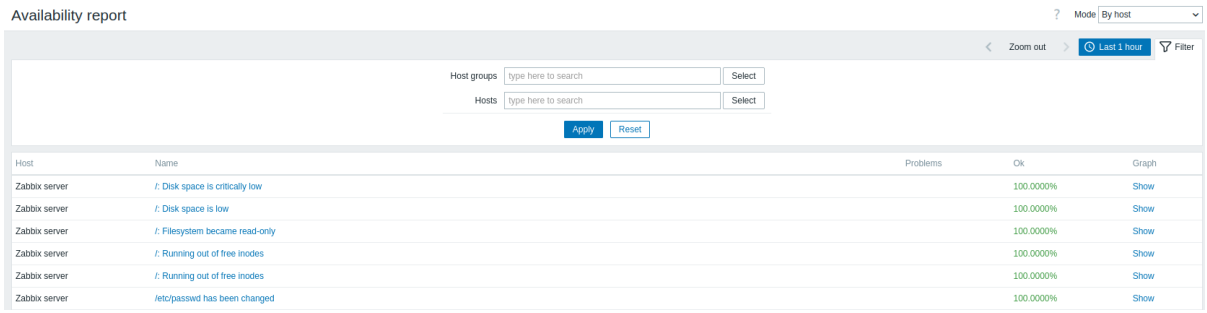
- Отметьте флажками отчеты, которые нужно обновить, в списке
- Нажмите нужную кнопку под списком, чтобы внести изменения (*Enable*, *Disable* или *Delete*)

Отчет о доступности

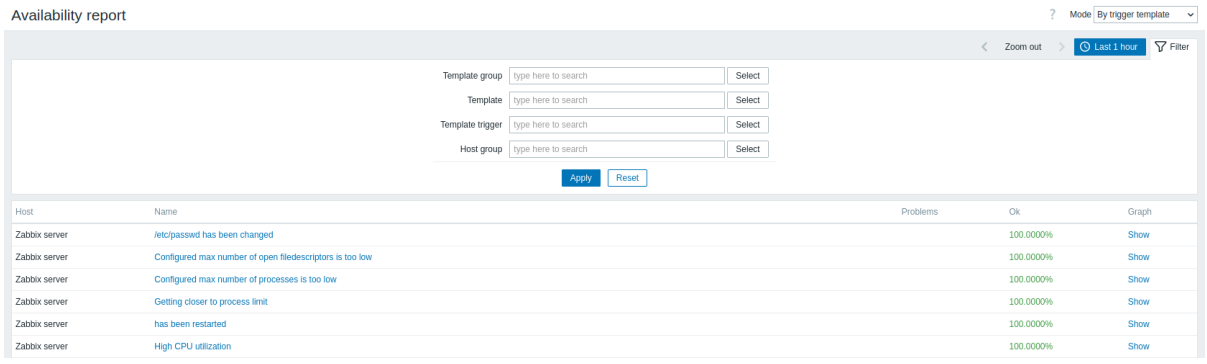
Обзор

В *Reports* > *Availability report* можно увидеть, какую долю времени каждый триггер находился в состоянии проблемы/OK.

Для каждого состояния отображается процент времени, что позволяет легко определить доступность различных элементов в вашей системе.



В раскрывающемся списке в правом верхнем углу можно выбрать режим отображения — показывать триггеры по узлам сети или триггеры, относящиеся к шаблону.



Имя триггера является ссылкой на последние события этого триггера.

Использование фильтра

Фильтр может помочь сузить количество отображаемых узлов сети и/или триггеров. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Фильтр расположен под названием раздела *Availability report*. Его можно открыть и свернуть, щелкнув вкладку *Filter* справа.

Фильтрация по шаблону триггера

В режиме *По шаблону триггера* результаты можно отфильтровать по одному или нескольким параметрам, перечисленным ниже.

Parameter	Description
Группа шаблонов	Фильтрация узлов сети по триггерам, унаследованным от шаблонов, принадлежащих выбранной группе шаблонов. При указании родительской группы шаблонов автоматически выбираются все вложенные группы шаблонов.
Шаблон	Фильтрация узлов сети по триггерам, унаследованным от выбранного шаблона, включая вложенные шаблоны. Если вложенный шаблон имеет собственные триггеры, эти триггеры отображаться не будут.
Триггер шаблона	Фильтрация узлов сети по выбранному триггеру. Другие триггеры отфильтрованных узлов сети отображаться не будут.
Группа узлов сети	Фильтрация узлов сети, принадлежащих выбранной группе узлов сети.

Фильтрация по узлу сети

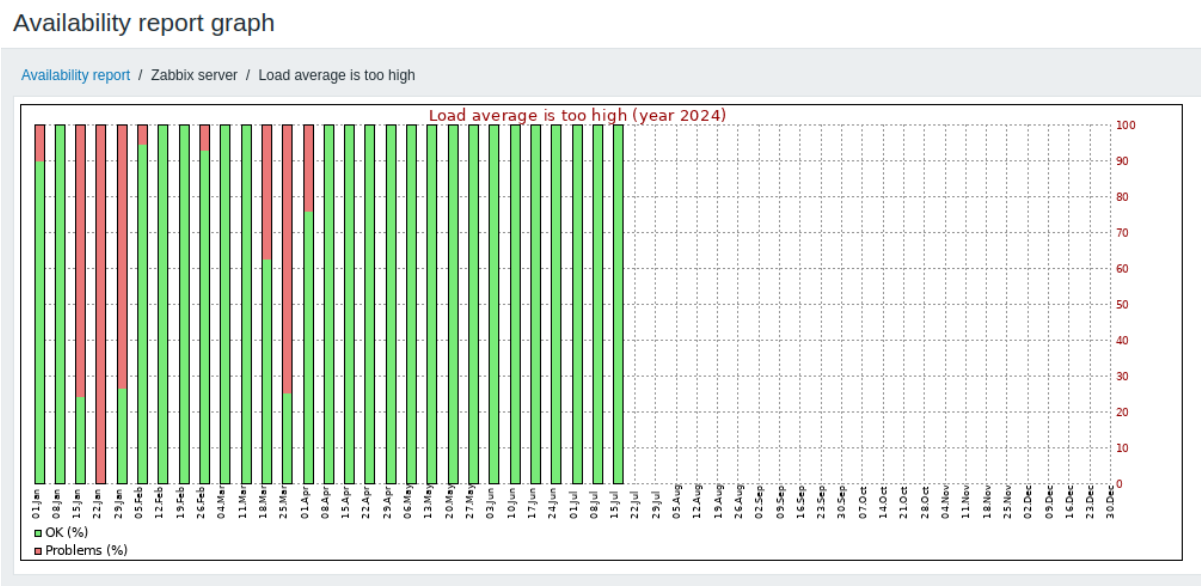
В режиме *По узлу сети* результаты можно фильтровать по узлу сети или группе узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети.

Выбор периода времени

Выбор периода времени позволяет одним щелчком выбрать часто используемые периоды времени. Выбор можно развернуть и свернуть, нажав вкладку *Период времени* рядом с фильтром.

При нажатии *Показать* в столбце *График* отображается информация о доступности в виде столбчатой диаграммы, где каждый столбец представляет прошедшую неделю текущего года.

Availability report graph



Зеленый цвет столбца обозначает время в состоянии «OK», а красный — время проблемы.

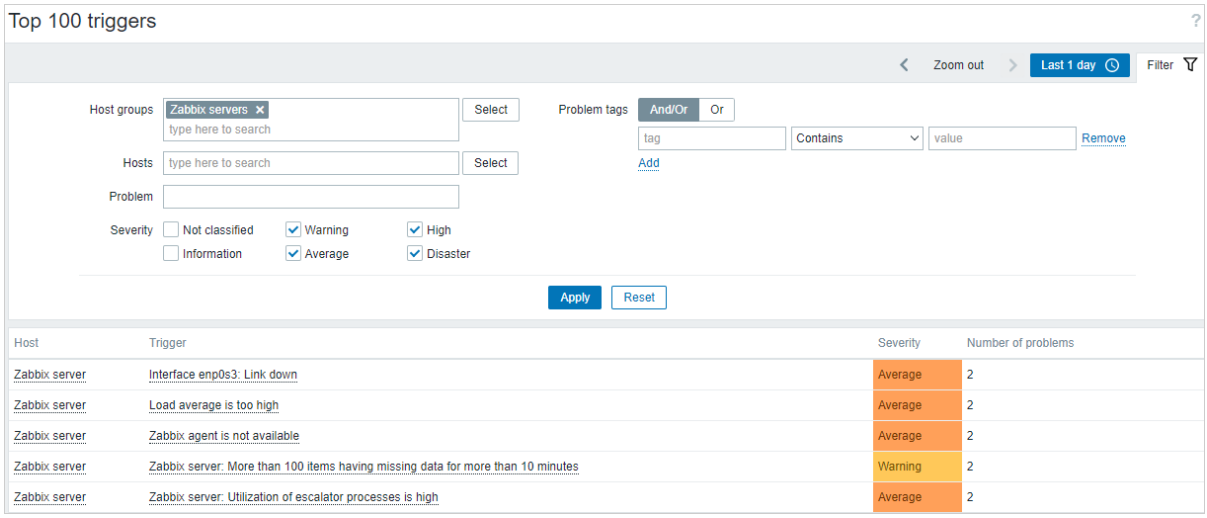
Влияние периодов обслуживания

Обслуживание не исключает время из отчета о доступности автоматически. Только обслуживание, настроенное как *No data collection*, останавливает опрос (поэтому проблемы не генерируются) и, следовательно, не изменяет процент доступности для затронутых триггеров.

Топ 100 триггеров

Обзор

В разделе *Отчеты* > *Топ-100 триггеров* можно просмотреть триггеры с наибольшим количеством проблем, обнаруженных за выбранный период.



Записи в столбцах узлов сети и триггеров являются ссылками и предоставляют несколько полезных возможностей:

- для узла сети — при нажатии на имя узла сети открывается меню узла сети
- для триггера — при нажатии на имя триггера открываются ссылки на последние события, простые графики для каждого элемента данных триггера, а также формы настройки самого триггера и каждого элемента данных триггера

Использование фильтра

С помощью фильтра можно отображать триггеры по группе узлов сети, узлу сети, имени проблемы, тегам или важности триггера. Указание родительской группы узлов сети автоматически выбирает все вложенные группы узлов сети. Для повышения производительности поиска данные ищутся с нераскрытыми макросами.

Фильтр расположен под названием раздела *Топ-100 триггеров*. Его можно открыть и свернуть, нажав вкладку *Фильтр* справа.

Выбор периода

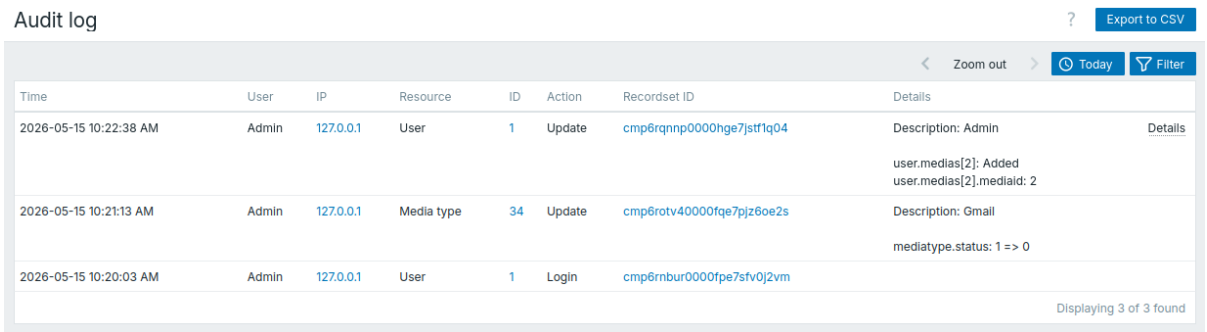
С помощью селектора периода рядом с фильтром можно одним щелчком мыши выбрать часто используемые периоды. Дополнительную информацию см. в разделе Выбор периода и узла сети.

Журнал аудита

Обзор

В разделе *Отчеты* > *Журнал аудита* можно просматривать записи о действиях пользователей и системы.

Note:
Чтобы записи аудита собирались и отображались, в разделе *Администрирование* > *Журнал аудита* должен быть отмечен флажок *Включить ведение журнала аудита*. Если этот параметр не включен, история действий не будет записываться в базу данных и не будет отображаться в журнале аудита.



Журнал аудита отображает следующие данные:

Column	Description
Time	Метка времени записи аудита.
User	Пользователь, выполнивший действие.
IP	IP, с которого было инициировано действие. При нажатии на гиперссылку записи журнала аудита будут отфильтрованы по этому IP.
Resource	Тип затронутого ресурса (*API token*, *Action*, *Authentication*, *Autoregistration* и т. д.).
ID	ID затронутого ресурса. При нажатии на гиперссылку записи журнала аудита будут отфильтрованы по этому ID ресурса.
Action	Тип действия (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	Общий ID для всех записей журнала аудита, созданных в результате одной и той же операции. Например, при связывании шаблона с узлом сети для каждой унаследованной сущности шаблона (элемент данных, триггер и т. д.) создается отдельная запись журнала аудита — все эти записи будут иметь один и тот же *Recordset ID*. При нажатии на гиперссылку записи журнала аудита будут отфильтрованы по этому *Recordset ID*.
Details	Описание ресурса и подробная информация о выполненном действии. Если запись содержит более двух строк, дополнительно будет отображаться ссылка *Details*. Нажмите эту ссылку, чтобы просмотреть полный список изменений.

Note:
Когда элемент данных типа trapper или элемент данных HTTP agent (с включенным trapping) получает данные, запись в журнале аудита будет добавлена только в том случае, если данные были отправлены с помощью метода API `history.push`, а не утилиты Zabbix sender.

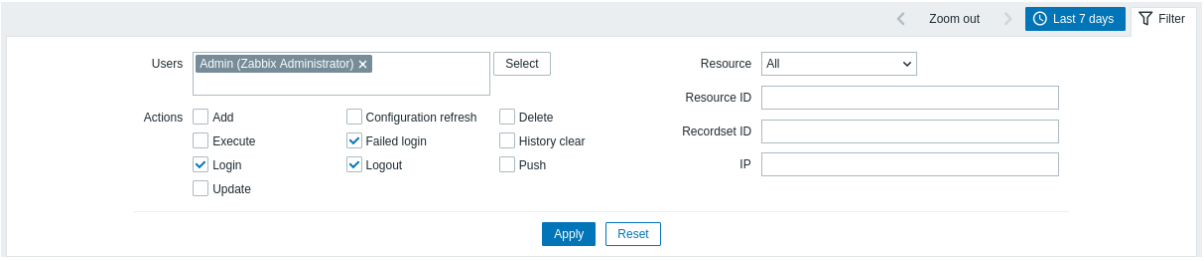
Кнопки

Кнопка в правом верхнем углу страницы предлагает следующую опцию:

Export to CSV	Экспортировать записи журнала аудита со всех страниц в CSV-файл. Если применен фильтр, будут экспортированы только отфильтрованные записи.

Использование фильтра

Фильтр расположен под панелью *Журнал аудита*. Его можно открыть и свернуть, щелкнув вкладку *Фильтр* в правом верхнем углу.



С помощью фильтра можно сузить список записей по пользователю, затронутому ресурсу, ID ресурса, выполненной операции (*ID набора записей*) и IP-адресу.

В зависимости от ресурса в фильтре можно выбрать одну или несколько конкретных действий.

Для повышения производительности поиска все данные ищутся с неразрешенными макросами.

Выбор периода времени

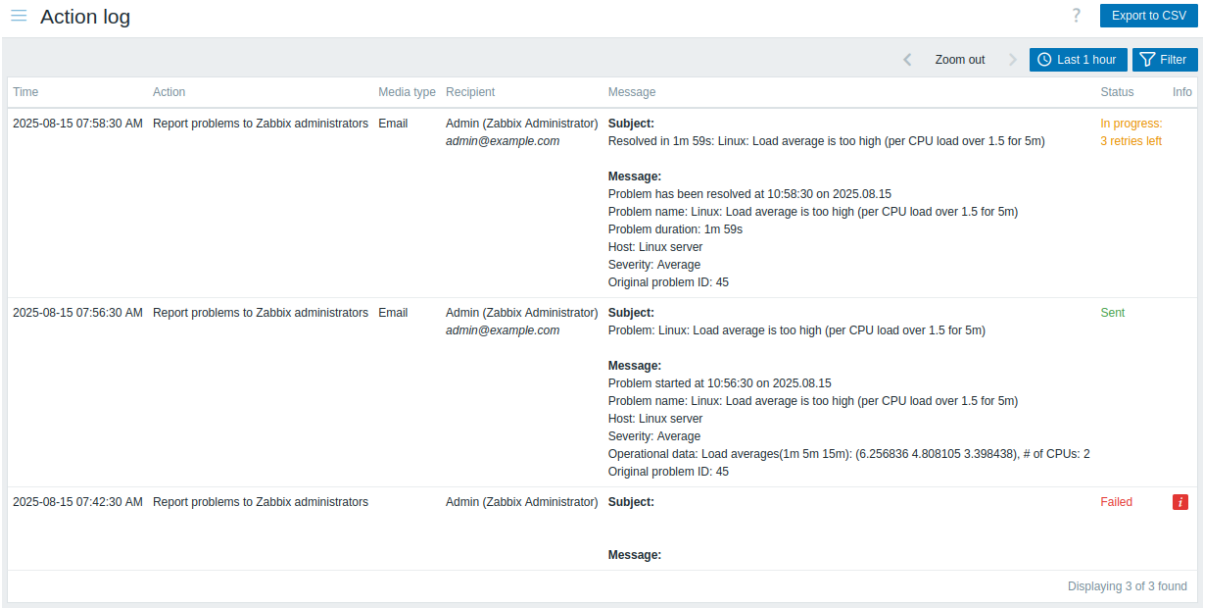
Выбор периода времени рядом с фильтром позволяет одним щелчком мыши выбрать часто используемые периоды.

Дополнительную информацию см. в разделе Выбор периода времени и узла сети.

Журнал действий

Обзор

В *Reports* > *Action log* можно просмотреть подробности операций (уведомления, удаленные команды), выполненных в рамках действия.



Отображаемые данные:

Column	Description
Time	Метка времени операции.
Action	Имя действия, вызвавшего операции.
Media type	Тип носителя (например, Email, Jira и т. д.), используемый для отправки уведомления. Для операций, которые выполняли удаленные команды, этот столбец будет пустым.
Recipient	Информация о получателе уведомления - имя пользователя, имя и фамилия (в скобках), а также дополнительная информация в зависимости от типа носителя (email, имя пользователя и т. д.). Для операций, которые выполняли удаленные команды, этот столбец будет пустым.
Message	Содержимое сообщения, удаленной команды или имя глобального скрипта. Удаленная команда отделяется от целевого узла сети двоеточием: `<host>:<command>`. Например, если удаленная команда была выполнена на Zabbix server, то информация будет иметь следующий формат: `Zabbix server:<command>`.
Status	Статус операции: *In progress* - операция отправки уведомления выполняется (также отображается оставшееся число попыток, которые сервер предпримет для отправки уведомления) *Sent* - уведомление отправлено *Executed* - удаленная команда выполнена *Failed* - операция не была завершена
Info	Информация об ошибке (если есть), относящаяся к выполнению операции.

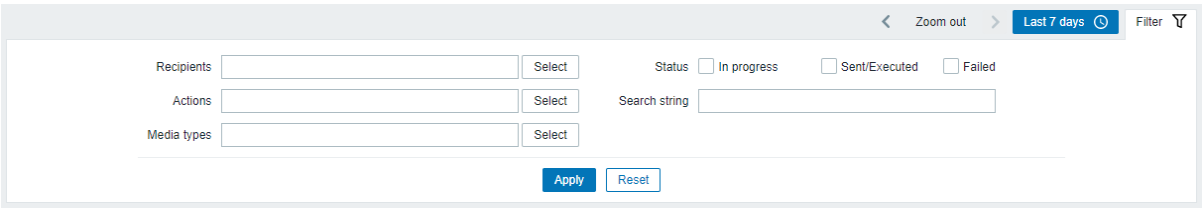
Кнопки

Кнопка в правом верхнем углу страницы предлагает следующую возможность:

Export to CSV	Экспортировать записи журнала действий с текущей страницы в CSV-файл. Если применен фильтр, будут экспортированы только отфильтрованные записи. В экспортированном CSV-файле столбцы ”Recipient” и ”Message” разделены на несколько столбцов - ”Recipient’s Zabbix username”, ”Recipient’s name”, ”Recipient’s surname”, ”Recipient”, а также ”Subject”, ”Message”, ”Command”.

Использование фильтра

Фильтр расположен под панелью *Журнал действий*. Его можно открыть и свернуть, щелкнув вкладку *Фильтр* в правом верхнем углу страницы.



Вы можете использовать фильтр, чтобы сузить список записей по получателям уведомлений, действиям, типам медиа, статусу или по содержимому сообщения/удаленной команды (*Строка поиска*). Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Выбор периода времени

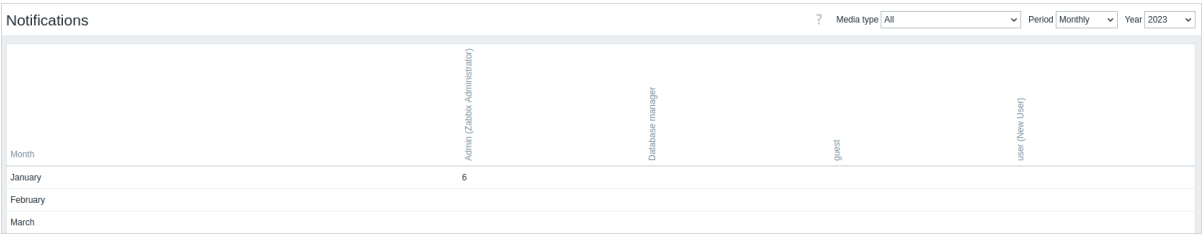
Выбор периода времени рядом с фильтром позволяет одним щелчком мыши выбрать часто используемые периоды. Дополнительную информацию см. в разделе Выбор периода времени и узла сети.

Уведомления

Обзор

В разделе *Reports* > *Notifications* отображается отчет о количестве уведомлений, отправленных каждому пользователю.

В раскрывающихся списках в правом верхнем углу можно выбрать тип медиа (или все), период (данные за каждый день/неделю/месяц/год) и год для отправленных уведомлений.



Каждый столбец отображает итоговые значения для одного системного пользователя.

Сбор данных

Обзор

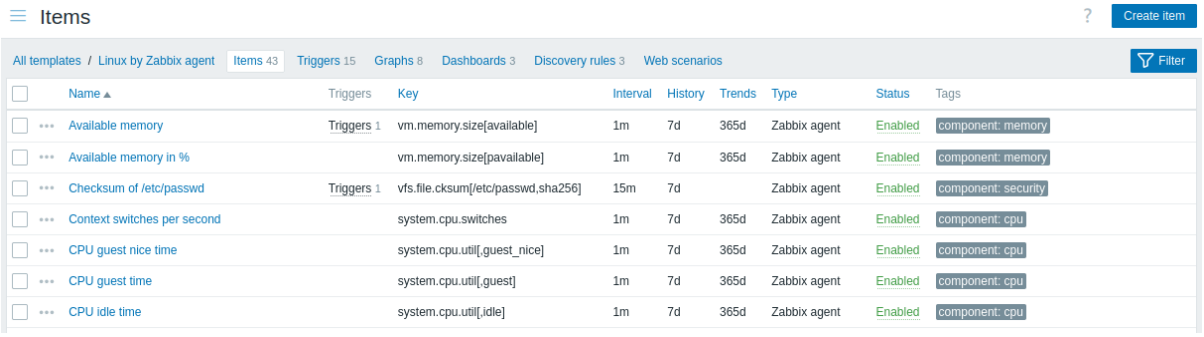
Это меню содержит разделы, связанные с настройкой сбора данных.

Элементы данных

Обзор

Список элементов данных для шаблона можно открыть в разделе *Сбор данных* > *Шаблоны*, нажав *Элементы данных* для соответствующего шаблона.

Отображается список существующих элементов данных.



Отображаемые данные:

Столбец	Описание
Меню элемента данных	Нажмите значок с тремя точками, чтобы открыть меню элемента данных.
Шаблон	Шаблон, которому принадлежит элемент данных. При нажатии имени шаблона открывается форма настройки шаблона. Этот столбец отображается только в том случае, если в фильтре выбрано несколько шаблонов или не выбран ни один шаблон.
Имя	Имя элемента данных, отображаемое в виде синей ссылки на сведения об элементе данных. При нажатии ссылки с именем элемента данных открывается форма настройки элемента данных. Если элемент данных унаследован от другого шаблона, перед его именем отображается имя шаблона в виде серой ссылки. При нажатии ссылки на шаблон открывается список элементов данных на уровне этого шаблона.
Триггеры	При наведении указателя мыши на пункт «Триггеры» отображается информационное окно со связанными с элементом данных триггерами. Количество триггеров отображается серым цветом.
Ключ	Отображается ключ элемента данных.
Интервал	Отображается частота проверки.
История	Отображается количество дней хранения истории данных элемента данных.
Тренды	Отображается количество дней хранения истории трендов элемента данных.
Тип	Отображается тип элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).
Состояние	Отображается состояние элемента данных: *Включен* или *Отключен*. Нажав состояние, его можно изменить с «Включен» на «Отключен» и наоборот.
Теги	Отображаются теги элемента данных. Можно отобразить до трех тегов (пар имя:значение). Если тегов больше, отображается ссылка «...», при наведении на которую можно просмотреть все теги.

Чтобы настроить новый элемент данных, нажмите кнопку *Создать элемент данных* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

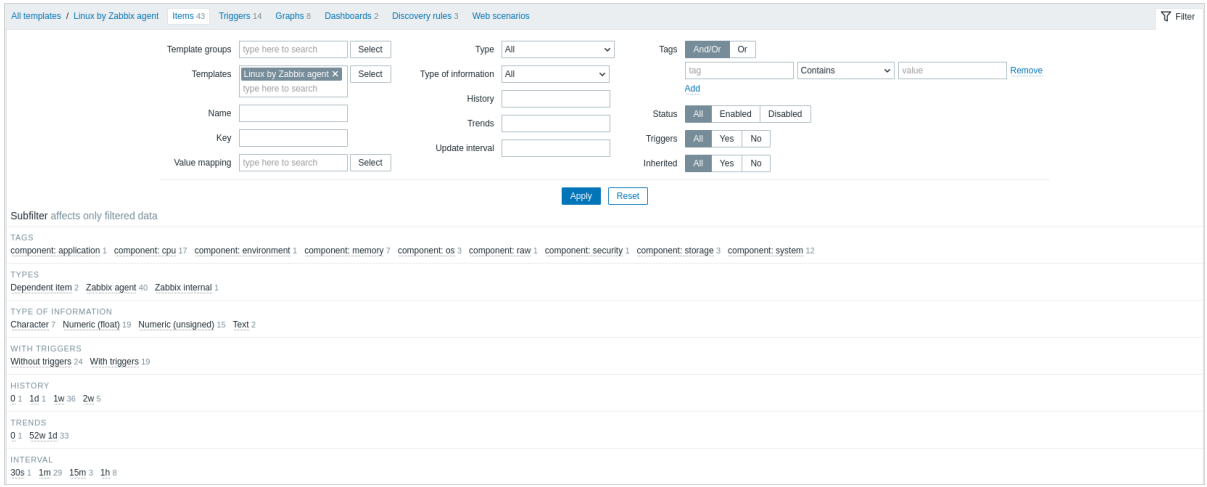
- *Enable* - изменить статус элемента данных на *Enabled*.
- *Disable* - изменить статус элемента данных на *Disabled*.
- *Copy* - скопировать элементы данных на другие узлы сети или шаблоны.
- *Mass update* - обновить несколько свойств для нескольких элементов данных одновременно.
- *Delete* - удалить элементы данных.

Чтобы использовать эти параметры, отметьте флажки перед нужными элементами данных, затем нажмите требуемую кнопку.

Использование фильтра

Список элементов данных может содержать большое количество записей. Используя фильтр, вы можете отфильтровать часть из них, чтобы быстро найти нужные элементы данных. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Фильтр* доступен в правом верхнем углу. При нажатии на него откроется фильтр, где можно задать нужные критерии фильтрации.

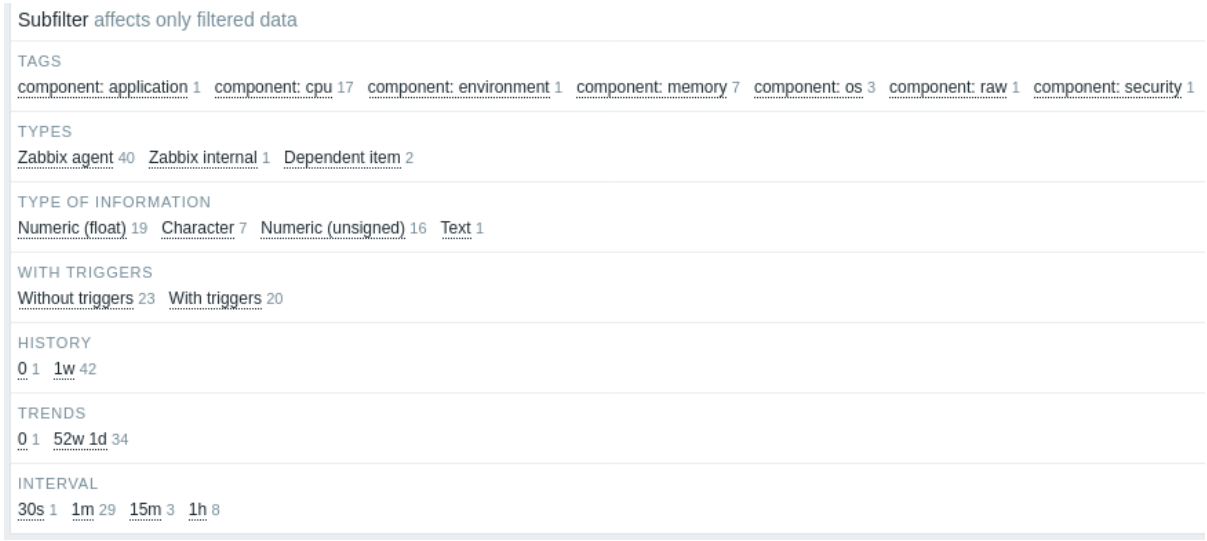


Parameter	Description
Template groups	Фильтрация по одной или нескольким группам шаблонов. При указании родительской группы шаблонов автоматически выбираются все вложенные группы.
Templates	Фильтрация по одному или нескольким шаблонам.
Name	Фильтрация по имени элемента данных.
Key	Фильтрация по ключу элемента данных.
Value mapping	Фильтрация по используемой карте значений. Этот параметр не отображается, если параметр *Templates* пуст.
Type	Фильтрация по типу элемента данных (агент Zabbix, агент SNMP и т. д.).
Type of information	Фильтрация по типу информации (беззнаковое число, число с плавающей точкой и т. д.).
History	Фильтрация по времени хранения истории элемента данных.
Trends	Фильтрация по времени хранения трендов элемента данных.
Update interval	Фильтрация по интервалу обновления элемента данных.
Tags	Укажите теги, чтобы ограничить количество отображаемых элементов данных. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия
Status	Фильтрация по статусу элемента данных - *Enabled* или *Disabled*.
Triggers	Фильтрация элементов данных с триггерами или без них.
Inherited	Фильтрация унаследованных или неунаследованных элементов данных из связанных шаблонов.

Использование подфильтра

Подфильтр позволяет дополнительно изменить фильтрацию, примененную основным фильтром.

Он содержит ссылки, по которым можно щелкнуть для быстрого доступа к связанным элементам данных. Элементы данных связаны общей сущностью: тегом, типом элемента данных, интервалом обновления элемента данных и т. д. При щелчке по сущности она выделяется серым фоном, а элементы данных немедленно фильтруются (нет необходимости нажимать *Применить* в основном фильтре). Щелчок по другой сущности добавляет ее в результаты фильтрации. Повторный щелчок по сущности отменяет фильтрацию по ней.



Подфильтры создаются на основе отфильтрованных данных, ограниченных 1000 записями. Если вы хотите видеть в подфильтре больше записей, увеличьте значение параметра *Ограничение результатов поиска и фильтрации* (в разделе *Администрирование* > *Общие* > *Графический интерфейс*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы всегда отображать актуальную информацию о доступных параметрах фильтрации и их количестве.

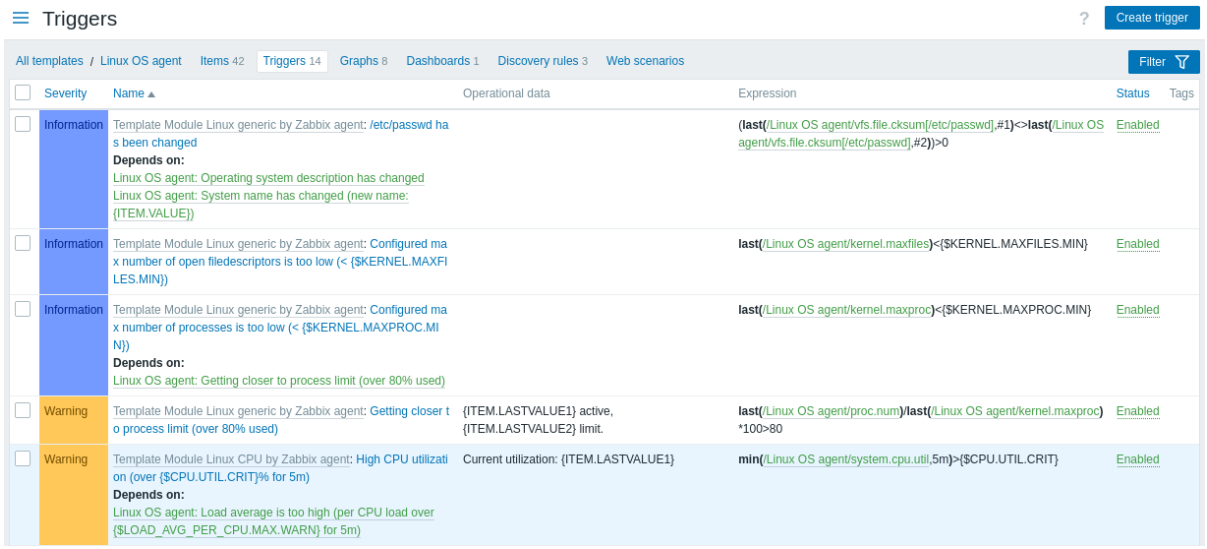
Количество отображаемых сущностей ограничено 100 по горизонтали. Если сущностей больше, в конце отображается значок с тремя точками; щелкнуть по нему нельзя.

Число рядом с каждой сущностью, по которой можно щелкнуть, указывает количество сгруппированных в ней элементов данных (на основе результатов основного фильтра). После щелчка по сущности рядом с другими доступными сущностями отображаются числа со знаком «плюс», указывающие, сколько элементов данных можно добавить к текущему выбору. Сущности без элементов данных не отображаются, если только они ранее не были выбраны в подфильтре.

Триггеры

Обзор

Список триггеров для шаблона можно открыть в разделе *Сбор данных* > *Шаблоны*, нажав *Триггеры* для соответствующего шаблона.



Отображаемые данные:

Столбец	Описание
Важность	Важность триггера отображается как с помощью названия, так и с помощью цвета фона ячейки.

Столбец	Описание
Шаблон	Шаблон, которому принадлежит триггер. При нажатии на название шаблона открывается форма настройки шаблона. Этот столбец отображается только в том случае, если в фильтре выбрано несколько шаблонов или не выбран ни один шаблон.
Название	Название триггера отображается в виде синей ссылки на сведения о триггере. При нажатии на ссылку с названием триггера открывается форма настройки триггера. Если триггер унаследован от другого шаблона, перед названием триггера отображается название шаблона в виде серой ссылки. При нажатии на ссылку с названием шаблона открывается список триггеров на уровне этого шаблона.
Операционные данные	Определение операционных данных триггера, содержащее произвольные строки и макросы, которые будут динамически обрабатываться в разделе *Мониторинг* > *Проблемы*.
Выражение	Отображается выражение триггера. Часть выражения, относящаяся к элементу данных шаблона, отображается в виде ссылки, ведущей к форме настройки элемента данных.
Состояние	Отображается состояние триггера — *Включен* или *Отключен*. Нажав на состояние, можно изменить его с «Включен» на «Отключен» и наоборот.
Теги	Если триггер содержит теги, в этом столбце отображаются имя и значение тега.

Чтобы настроить новый триггер, нажмите кнопку *Создать триггер* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

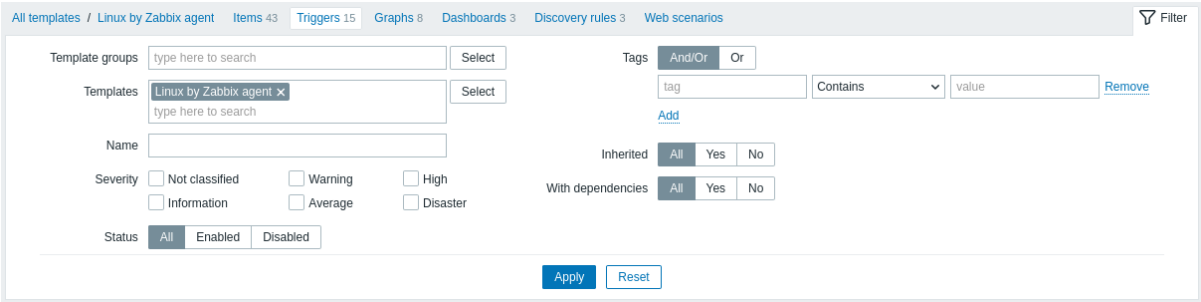
- *Enable* - изменить статус триггера на *Enabled*
- *Disable* - изменить статус триггера на *Disabled*
- *Copy* - скопировать триггеры на другие узлы сети или шаблоны
- *Mass update* - обновить несколько свойств для нескольких триггеров одновременно
- *Delete* - удалить триггеры

Чтобы использовать эти параметры, отметьте флажки перед нужными триггерами, затем нажмите требуемую кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те триггеры, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Фильтр* доступен в правом верхнем углу. При нажатии на него откроется фильтр, где можно указать нужные критерии отбора.



Параметр	Описание
Группы шаблонов	Фильтрация по одной или нескольким группам шаблонов. При указании родительской группы шаблонов автоматически выбираются все вложенные группы.
Шаблоны	Фильтрация по одному или нескольким шаблонам. Если выше уже выбраны группы шаблонов, выбор шаблонов ограничивается этими группами.
Имя	Фильтрация по имени триггера.
Серьезность	Выберите, чтобы фильтровать по одной или нескольким степеням серьезности триггера.
Статус	Фильтрация по статусу активации триггера (Включен/Выключен).

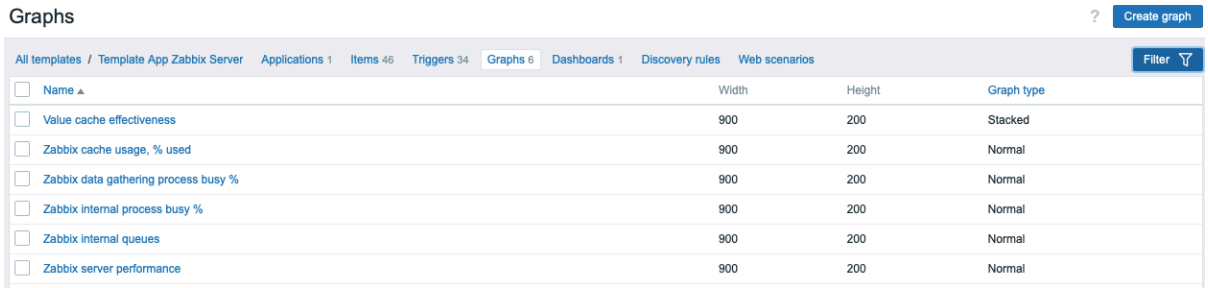
Параметр	Описание
Теги	Фильтрация по имени и значению тега триггера. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия В полях имени тега и значения тега поддерживаются макросы и функции макросов.
Наследованные	Фильтрация триггеров, унаследованных (или не унаследованных) от связанных шаблонов.
С зависимостями	Фильтрация триггеров с зависимостями (или без зависимостей).

Графики

Обзор

Список пользовательских графиков для шаблона можно открыть в разделе *Сбор данных* > *Шаблоны*, нажав *Графики* для соответствующего шаблона.

Отображается список существующих графиков.



Отображаемые данные:

Столбец	Описание
Шаблон	Шаблон, которому принадлежит график. При нажатии на имя шаблона открывается форма настройки шаблона. Этот столбец отображается только в том случае, если в фильтре выбрано несколько шаблонов или не выбран ни один шаблон.
Имя	Имя пользовательского графика, отображаемое в виде синей ссылки на сведения о графике. При нажатии на ссылку с именем графика открывается форма настройки графика. Если график унаследован от другого шаблона, перед именем графика отображается имя шаблона в виде серой ссылки. При нажатии на ссылку с именем шаблона открывается список графиков на уровне этого шаблона.
Ширина	Отображается ширина графика.
Высота	Отображается высота графика.
Тип графика	Отображается тип графика: *Обычный*, *С накоплением*, *Круговая диаграмма* или *Взрывная диаграмма*.

Чтобы настроить новый график, нажмите кнопку *Создать график* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Copy* - копировать графики на другие узлы сети или шаблоны
- *Delete* - удалить графики

Чтобы использовать эти параметры, отметьте флажки перед соответствующими графиками, затем нажмите нужную кнопку.

Использование фильтра

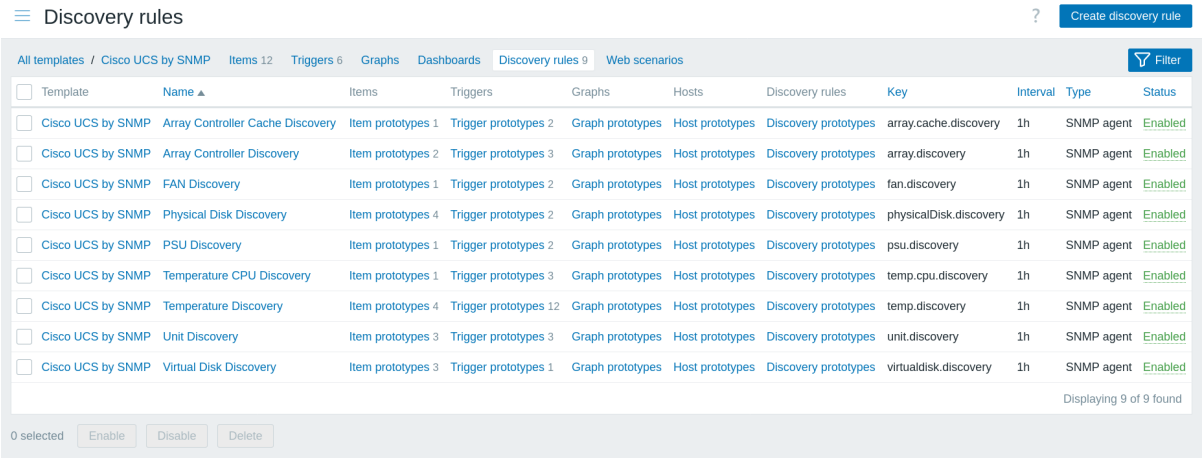
Вы можете фильтровать графики по группе шаблонов и шаблону. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Правила обнаружения

Обзор

Список правил низкоуровневого обнаружения для шаблона можно открыть в разделе *Сбор данных* > *Шаблоны*, нажав *Обнаружение* для соответствующего шаблона.

Отображается список существующих правил низкоуровневого обнаружения. Также можно просмотреть все правила обнаружения независимо от шаблона или все правила обнаружения определенной группы шаблонов, изменив настройки фильтра.



Отображаемые данные:

Столбец	Описание
Шаблон	Шаблон, которому принадлежит правило обнаружения. При нажатии имени шаблона открывается форма настройки шаблона.
Имя	Имя правила, отображаемое в виде синей ссылки. При нажатии имени правила открывается форма настройки правила низкоуровневого обнаружения. Если правило обнаружения унаследовано от другого шаблона, перед именем правила отображается имя шаблона в виде серой ссылки. При нажатии ссылки на шаблон открывается список правил обнаружения на уровне этого шаблона.
Элементы данных	Отображается ссылка на список прототипов элементов данных. Количество существующих прототипов элементов данных отображается серым цветом.
Триггеры	Отображается ссылка на список прототипов триггеров. Количество существующих прототипов триггеров отображается серым цветом.
Графики	Отображается ссылка на список прототипов графиков. Количество существующих прототипов графиков отображается серым цветом.
Узлы сети	Отображается ссылка на список прототипов узлов сети. Количество существующих прототипов узлов сети отображается серым цветом.
Ключ	Отображается ключ элемента данных, используемый для обнаружения.
Интервал	Отображается периодичность выполнения обнаружения.
Тип	Отображается тип элемента данных, используемый для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Состояние	Отображается состояние правила обнаружения: *Включено* или *Отключено*. Нажав состояние, можно изменить его с «Включено» на «Отключено» и наоборот.

Чтобы настроить новое правило низкоуровневого обнаружения, нажмите кнопку *Создать правило обнаружения* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

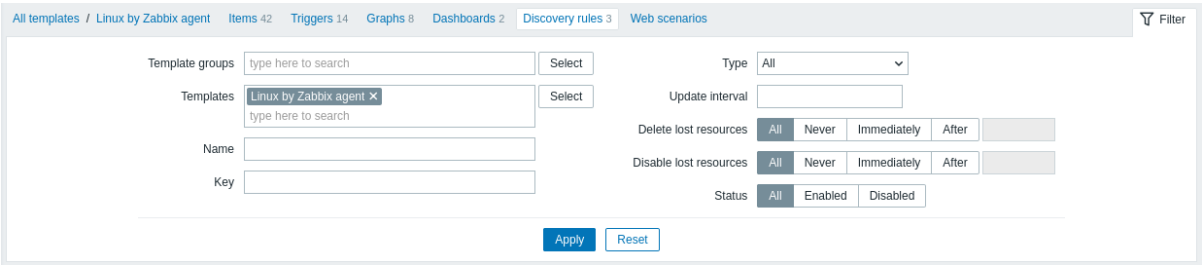
- *Включить* - изменить статус правила обнаружения низкого уровня на *Включено*
- *Отключить* - изменить статус правила обнаружения низкого уровня на *Отключено*
- *Удалить* - удалить правила обнаружения низкого уровня

Чтобы использовать эти параметры, отметьте флажки перед соответствующими правилами обнаружения, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила обнаружения, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Фильтр* доступен в правом верхнем углу. При нажатии на него откроется фильтр, где можно указать нужные критерии фильтрации, такие как шаблон, имя правила обнаружения, ключ элемента данных, тип элемента данных и т. д.



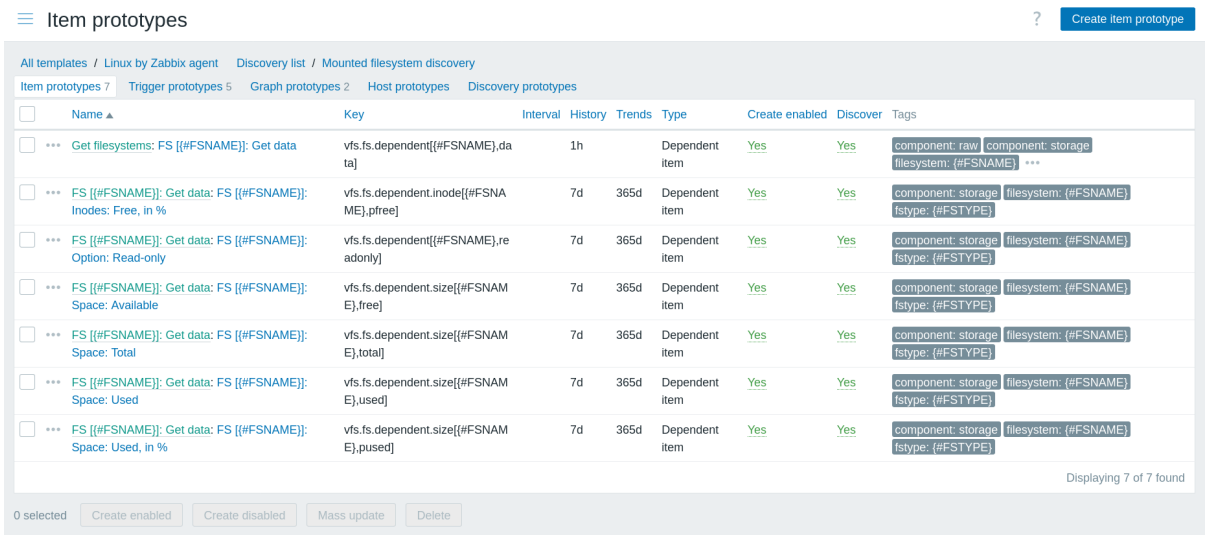
Parameter	Description
Template groups	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы.
Templates	Фильтрация по одному или нескольким шаблонам.
Name	Фильтрация по имени правила обнаружения.
Key	Фильтрация по ключу элемента данных обнаружения.
Type	Фильтрация по типу элемента данных обнаружения.
Update interval	Фильтрация по интервалу обновления. Недоступно для Zabbix trapper и зависимых элементов данных.
Delete lost resources	Фильтрация по периоду *Delete lost resources*.
Disable lost resources	Фильтрация по периоду *Disable lost resources*.
Status	Фильтрация по статусу правила обнаружения (All/Enabled/Disabled).

Прототипы элементов данных

Обзор

В этом разделе отображаются настроенные прототипы элементов данных правила обнаружения низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы элементов данных станут основой для создания реальных элементов данных узла сети во время низкоуровневого обнаружения.



Отображаемые данные:

Column	Description
Name	Отображается имя прототипа элемента данных в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа элемента данных. Если прототип элемента данных принадлежит связанному шаблону, перед именем прототипа элемента данных отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов элементов данных на уровне связанного шаблона. Если прототип элемента данных имеет тип «Зависимый элемент данных», перед его именем зеленым цветом отображается имя главного элемента данных (как в случае прототипов элементов данных на уровне узла сети).
Key	Отображается ключ прототипа элемента данных.
Interval	Отображается частота проверки.
History	Отображается количество дней хранения истории данных элемента данных.
Trends	Отображается количество дней хранения истории трендов элемента данных.
Type	Отображается тип прототипа элемента данных (агент Zabbix, агент SNMP, простая проверка и т. д.).
Create enabled	Создание элемента данных на основе этого прототипа: **Yes** — включено **No** — отключено. Переключаться между значениями «Yes» и «No» можно, нажимая на них.
Discover	Обнаружение элемента данных на основе этого прототипа: **Yes** — обнаруживать **No** — не обнаруживать. Переключаться между значениями «Yes» и «No» можно, нажимая на них.
Tags	Отображаются теги прототипа элемента данных.

Чтобы настроить новый прототип элемента данных, нажмите кнопку *Create item prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Создать включенными* - создать эти элементы данных как *Включенные*
- *Создать отключенными* - создать эти элементы данных как *Отключенные*
- *Массовое обновление* - массово обновить эти прототипы элементов данных
- *Удалить* - удалить эти прототипы элементов данных

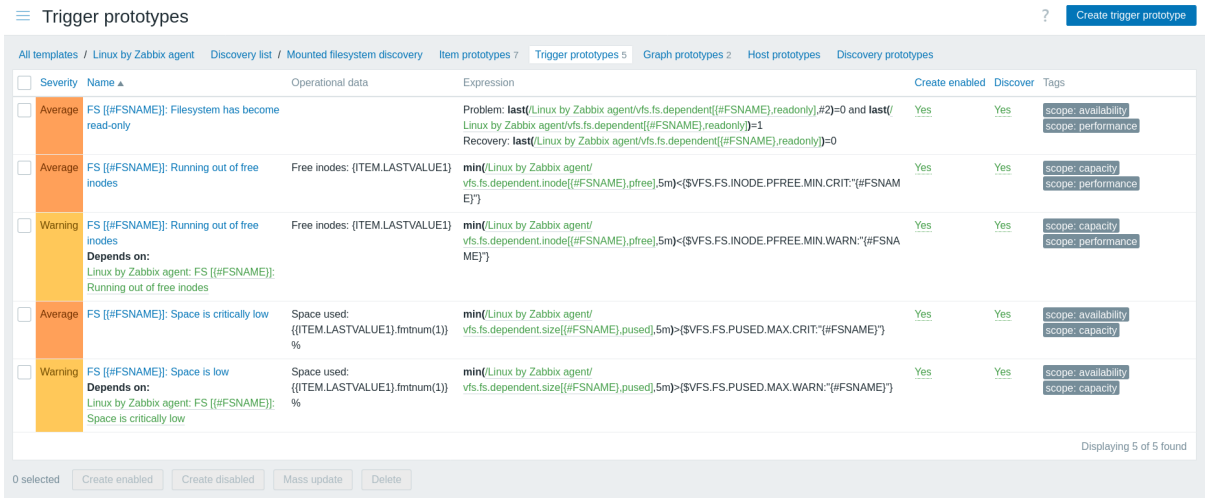
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами элементов данных, затем нажмите нужную кнопку.

Прототипы триггеров

Обзор

В этом разделе отображаются настроенные прототипы триггеров правила обнаружения низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы триггеров станут основой для создания реальных триггеров узла сети во время низкоуровневого обнаружения.



Отображаемые данные:

Column	Description
Severity	Степень серьезности триггера отображается как названием, так и цветом фона ячейки.
Name	Имя прототипа триггера отображается в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа триггера. Если прототип триггера принадлежит связанному шаблону, перед именем триггера отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов триггеров на уровне связанного шаблона.
Expression	Отображается выражение триггера. Часть выражения, относящаяся к элементу данных шаблона, отображается в виде ссылки, ведущей к форме настройки элемента данных.
Operational data	Отображается формат рабочих данных триггера, содержащий произвольные строки и макросы, которые будут динамически обрабатываться в разделе *Мониторинг > Проблемы*.
Create enabled	Создание триггера на основе этого прототипа: **Yes** - включено **No** - отключено. Переключаться между значениями «Yes» и «No» можно, нажимая на них.
Discover	Обнаружение триггера на основе этого прототипа: **Yes** - обнаруживать **No** - не обнаруживать. Переключаться между значениями «Yes» и «No» можно, нажимая на них.
Tags	Отображаются теги прототипа триггера.

Чтобы настроить новый прототип триггера, нажмите кнопку *Create trigger prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Создать включенными* - создать эти триггеры как *Включенные*
- *Создать отключенными* - создать эти триггеры как *Отключенные*
- *Массовое обновление* - массово обновить эти прототипы триггеров
- *Удалить* - удалить эти прототипы триггеров

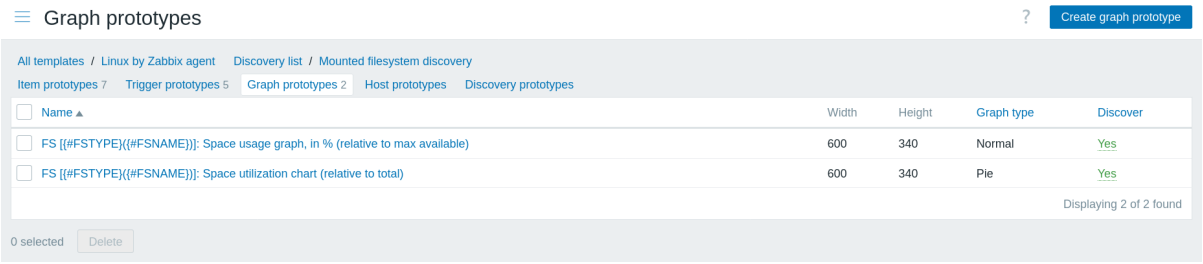
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами триггеров, затем нажмите нужную кнопку.

Прототипы графиков

Обзор

В этом разделе отображаются настроенные прототипы графиков правила обнаружения низкоуровневых объектов на шаблоне.

Если шаблон связан с узлом сети, прототипы графиков станут основой для создания реальных графиков узла сети в процессе обнаружения низкоуровневых объектов.



Отображаемые данные:

Column	Description
Name	Имя прототипа графика, отображаемое в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа графика. Если прототип графика принадлежит связанному шаблону, перед именем графика отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов графиков на уровне связанного шаблона.
Width	Отображается ширина прототипа графика.
Height	Отображается высота прототипа графика.
Type	Отображается тип прототипа графика: *Normal*, *Stacked*, *Pie* или *Exploded*.
Discover	Обнаруживать график на основе этого прототипа: **Yes** — обнаруживать **No** — не обнаруживать. Переключаться между значениями «Yes» и «No» можно, нажимая на них.

Чтобы настроить новый прототип графика, нажмите кнопку *Create graph prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Удалить* - удалить эти прототипы графиков

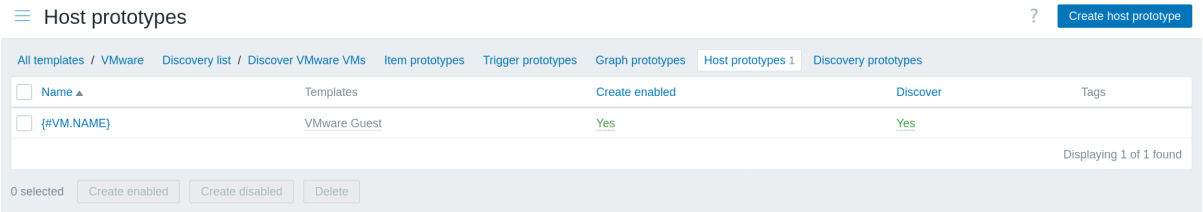
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами графиков, затем нажмите нужную кнопку.

Прототипы узлов сети

Обзор

В этом разделе отображаются настроенные прототипы узлов сети правила обнаружения низкого уровня в шаблоне.

Если шаблон связан с узлом сети, прототипы узлов сети будут использоваться для создания реальных узлов сети во время обнаружения низкого уровня.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа узла сети, отображаемое в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа узла сети. Если прототип узла сети принадлежит связанному шаблону, перед именем узла сети отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов узлов сети на уровне связанного шаблона.
Шаблоны	Отображаются шаблоны прототипа узла сети.
Создание включено	Создавать узел сети на основе этого прототипа: **Да** - включено **Нет** - отключено. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Обнаруживать	Обнаруживать узел сети на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Теги	Отображаются теги прототипа узла сети.

Чтобы настроить новый прототип узла сети, нажмите кнопку *Создать прототип узла сети* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Создать включенными* - создать эти узлы сети как *Включенные*
- *Создать отключенными* - создать эти узлы сети как *Отключенные*
- *Удалить* - удалить эти прототипы узлов сети

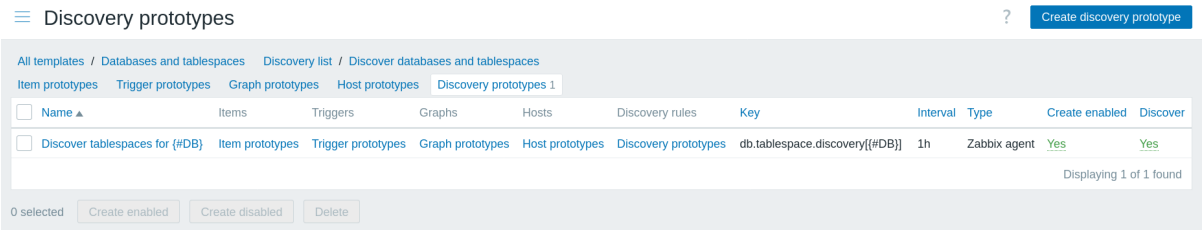
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами узлов сети, затем нажмите нужную кнопку.

Прототипы обнаружения

Обзор

В этом разделе отображаются настроенные прототипы обнаружения правила обнаружения низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы обнаружения станут основой для создания реальных прототипов обнаружения родительского правила обнаружения.



Отображаемые данные:

Столбец	Описание
Name	Имя прототипа обнаружения, отображаемое в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа обнаружения. Если прототип обнаружения принадлежит связанному шаблону, перед именем прототипа обнаружения отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов обнаружения на уровне связанного шаблона.
Items	Отображается ссылка на список прототипов элементов данных. Количество существующих прототипов элементов данных отображается серым цветом.
Triggers	Отображается ссылка на список прототипов триггеров. Количество существующих прототипов триггеров отображается серым цветом.
Graphs	Отображается ссылка на список прототипов графиков. Количество существующих прототипов графиков отображается серым цветом.
Hosts	Отображается ссылка на список прототипов узлов сети. Количество существующих прототипов узлов сети отображается серым цветом.

Столбец	Описание
Discovery rules	Отображается ссылка на список прототипов обнаружения. Количество существующих прототипов обнаружения отображается серым цветом.
Key	Отображается ключ элемента данных, используемый для обнаружения.
Interval	Отображается частота выполнения обнаружения.
Type	Отображается тип элемента данных, используемый для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Create enabled	Отображается состояние создания прототипа обнаружения: создание включено (*Yes*) или создание отключено (*No*). Нажав на состояние, его можно изменить с Yes на No и обратно.
Discover	Отображается состояние обнаружения прототипа обнаружения: обнаруживать (*Yes*) или не обнаруживать (*No*). Нажав на состояние, его можно изменить с Yes на No и обратно.

Чтобы настроить новый прототип обнаружения, нажмите кнопку *Create discovery prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Создать включенными* - создать эти прототипы обнаружения как *Включенные*
- *Создать отключенными* - создать эти прототипы обнаружения как *Отключенные*
- *Удалить* - удалить эти прототипы обнаружения

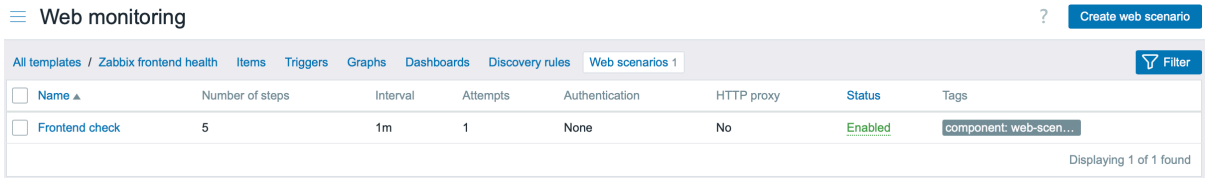
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами обнаружения, затем нажмите нужную кнопку.

Веб-сценарии

Обзор

Список веб-сценариев для шаблона можно открыть в разделе *Сбор данных* > *Шаблоны*, нажав *Веб* для соответствующего шаблона.

Отображается список существующих веб-сценариев.



Отображаемые данные:

Столбец	Описание
Имя	Имя веб-сценария. При нажатии на имя веб-сценария открывается форма настройки веб-сценария. Если веб-сценарий унаследован от другого шаблона, перед именем веб-сценария отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список веб-сценариев на уровне этого шаблона.
Количество шагов	Количество шагов в сценарии.
Интервал обновления	Частота выполнения сценария.
Попытки	Количество попыток выполнения шагов веб-сценария.
Аутентификация	Отображается метод аутентификации: Basic, NTLM, Kerberos, Digest или None.
HTTP-прокси	Отображается HTTP-прокси или «Нет», если он не используется.
Состояние	Отображается состояние веб-сценария: *Включен* или *Отключен*. Нажав на состояние, его можно изменить.
Теги	Отображаются теги веб-сценария. Можно отобразить до трех тегов (пар имя:значение). Если тегов больше, отображается ссылка «...», при наведении указателя мыши на которую можно просмотреть все теги.

Чтобы настроить новый веб-сценарий, нажмите кнопку *Создать веб-сценарий* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Enable* - изменить статус сценария на *Enabled*
- *Disable* - изменить статус сценария на *Disabled*
- *Delete* - удалить веб-сценарии

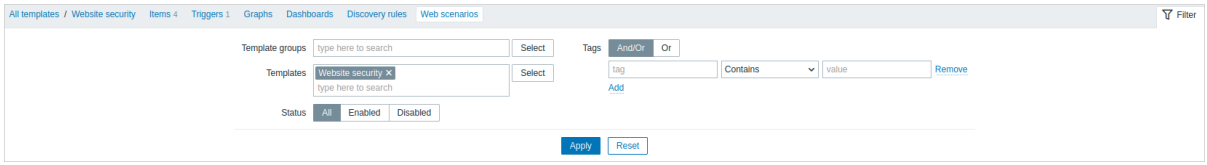
Чтобы использовать эти параметры, отметьте флажки перед соответствующими веб-сценариями, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те сценарии, которые вас интересуют.

Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком веб-сценариев. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать сценарии по группе шаблонов, шаблону, статусу и тегам.

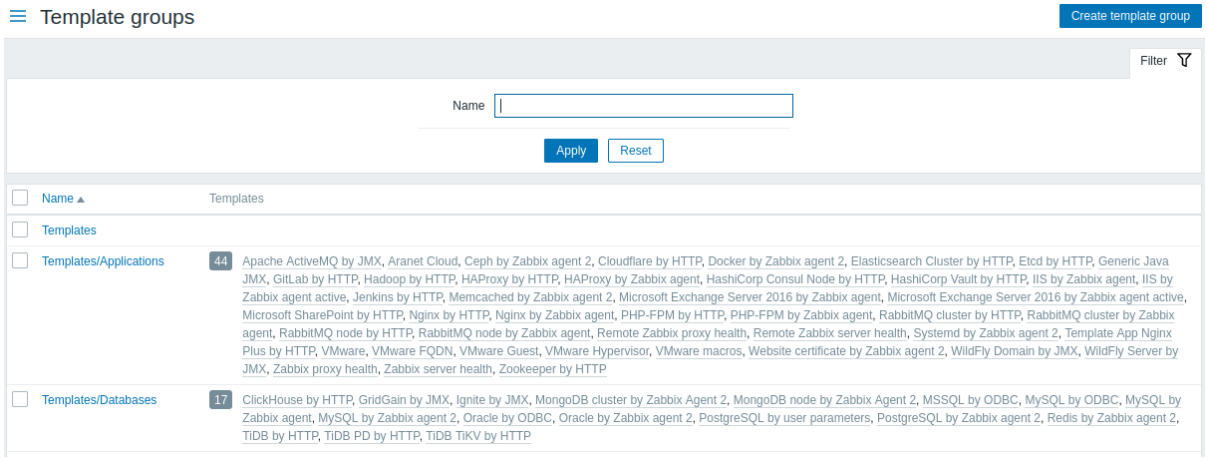


Группы шаблонов

Обзор

В разделе *Сбор данных* > *Группы шаблонов* пользователи могут настраивать и поддерживать группы шаблонов.

Отображается список существующих групп шаблонов с подробной информацией о них. Вы можете выполнять поиск и фильтровать группы шаблонов по имени.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы шаблонов. При нажатии на имя группы открывается форма настройки группы шаблонов.
Шаблоны	Количество шаблонов в группе (отображается серым цветом), за которым следует список участников группы. При нажатии на имя шаблона открывается форма настройки шаблона. При нажатии на число открывается список шаблонов в этой группе.

Параметры массового редактирования

Чтобы удалить несколько групп шаблонов одновременно, отметьте флажки напротив соответствующих групп, затем нажмите кнопку Delete под списком.

Использование фильтра

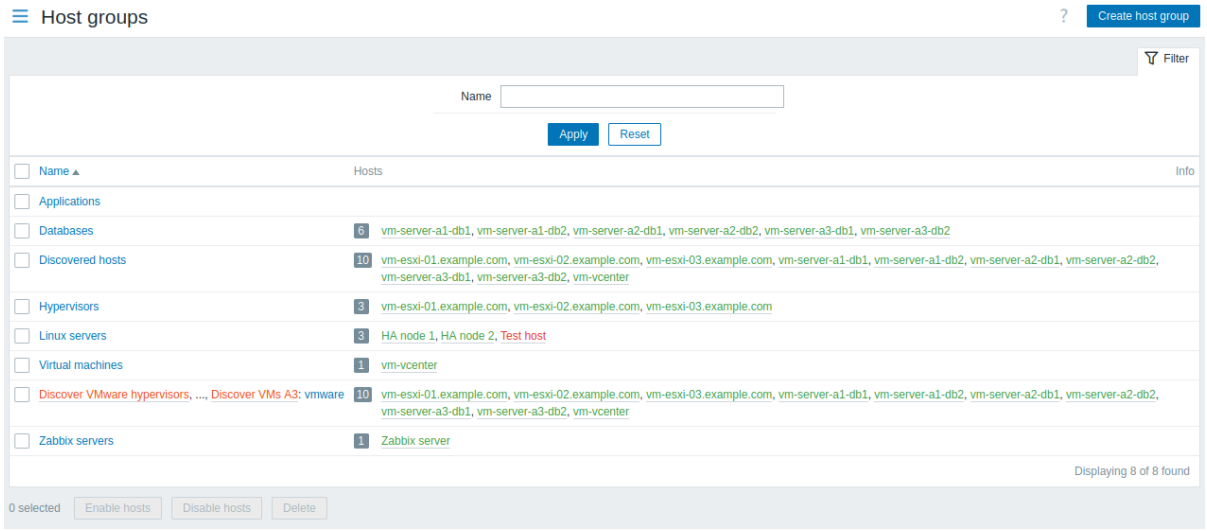
Вы можете использовать фильтр, чтобы отображать только те группы шаблонов, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Группы узлов сети

Обзор

В разделе *Сбор данных* > *Группы узлов сети* пользователи могут настраивать и обслуживать группы узлов сети.

Отображается список существующих групп узлов сети с подробной информацией о них. Вы можете выполнять поиск и фильтрацию групп узлов сети по имени.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы узлов сети. При нажатии на имя группы открывается форма настройки группы узлов сети. Обнаруженные группы узлов сети отображаются с именами правил низкоуровневого обнаружения в качестве префикса. При нажатии на имя правила НУО открывается форма настройки прототипа узла сети. Обратите внимание, что обнаруженные группы узлов сети удаляются после удаления всех правил НУО, которые их обнаружили.
Узлы сети	Количество узлов сети в группе (отображается серым цветом), за которым следует список участников группы. При нажатии на имя узла сети открывается форма настройки узла сети. При нажатии на число в общем списке узлов сети отображаются только узлы сети, принадлежащие этой группе.
Информация	Отображается информация об ошибках (если таковые имеются), связанных с группой узлов сети.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Включить узлы сети* - изменить статус всех узлов сети в группе на ”Контролируется”
- *Отключить узлы сети* - изменить статус всех узлов сети в группе на ”Не контролируется”
- *Удалить* - удалить группы узлов сети

Чтобы использовать эти параметры, отметьте флажки перед соответствующими группами узлов сети, затем нажмите нужную кнопку.

Использование фильтра

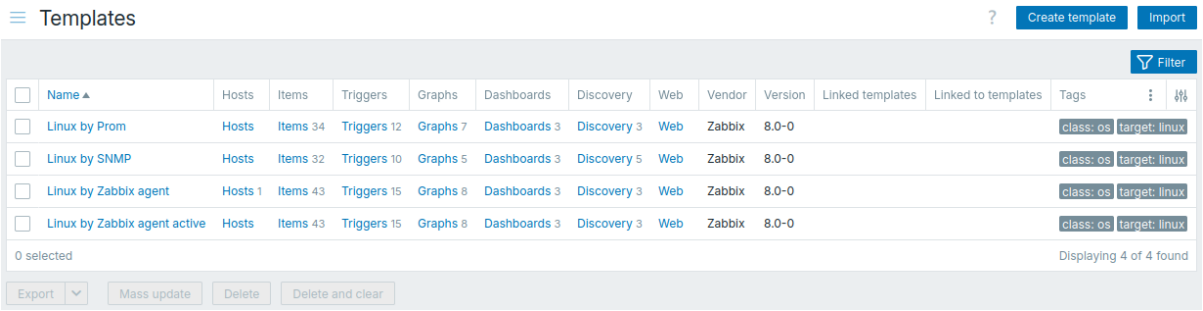
Вы можете использовать фильтр, чтобы отображать только те группы узлов сети, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Шаблоны

Обзор

В разделе *Сбор данных* > *Шаблоны* пользователи могут настраивать и обслуживать шаблоны.

Отображается список существующих шаблонов с подробной информацией о них.



Подробная информация о шаблонах отображается в настраиваемой таблице. Кнопка *Настроить таблицу* в заголовке таблицы позволяет отображать, скрывать или изменять порядок столбцов, а также восстанавливать макет по умолчанию. Размер каждого столбца также можно изменить, перетащив его правую границу; для сброса размера дважды щелкните границу. Все изменения применяются немедленно и всегда сохраняются для текущего пользователя.

Доступны следующие столбцы:

Столбец	Описание
Имя	Имя шаблона. При нажатии на имя шаблона открывается форма настройки шаблона.
Узлы сети	Количество редактируемых узлов сети, с которыми связан шаблон; узлы сети, доступные только для чтения, не учитываются. При нажатии *Узлы сети* открывается список узлов сети, отфильтрованный по узлам сети, связанным с шаблоном.
Сущности (элементы данных, триггеры, графики, панели мониторинга, обнаружение, веб)	Количество соответствующих сущностей в шаблоне (отображается серым цветом). При нажатии на имя сущности в полном списке этой сущности отображаются только сущности, принадлежащие шаблону.
Связанные шаблоны *Шаблоны, с которыми связан*	Шаблоны, которые связаны с этим шаблоном. Шаблоны, с которыми этот шаблон связан.
Производитель, *Версия*	Производитель и версия шаблона; отображаются, если конфигурация шаблона содержит такую информацию, и только для готовых шаблонов, импортированных шаблонов или шаблонов, измененных через Template API. Для готовых шаблонов версия отображается следующим образом: основная версия Zabbix, разделитель (”-”), номер редакции (увеличивается с каждой новой версией шаблона и сбрасывается при каждой основной версии Zabbix). Например: 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Теги	Отображаются теги шаблона без раскрытия макросов. При наведении курсора на значок с тремя точками или нажатии на него после отображаемых тегов открывается всплывающее окно со всеми тегами. Кнопка *Параметры* в заголовке столбца позволяет задать количество отображаемых тегов (1–3), выбрать способ отображения имен тегов (полностью, сокращенно или не отображать), задать приоритет отображения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Значение тега	Отображаются значения тегов шаблона без раскрытия макросов. Кнопка *Параметры* в заголовке столбца позволяет переименовать столбец, выбрать отображаемые значения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Пользовательский текст	Отображается пользовательский текст, значения макросов или их сочетание. Кнопка *Параметры* в заголовке столбца позволяет переименовать столбец, ввести пользовательский текст или макросы либо дублировать столбец (дублированные столбцы можно переименовать или удалить). Поддерживаемые макросы: `{HOST.ID}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.DESCRIPTION}` и пользовательские макросы.

Чтобы настроить новый шаблон, нажмите кнопку *Создать шаблон* в правом верхнем углу.

Чтобы импортировать шаблон из файла YAML, XML или JSON, нажмите кнопку *Импорт* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

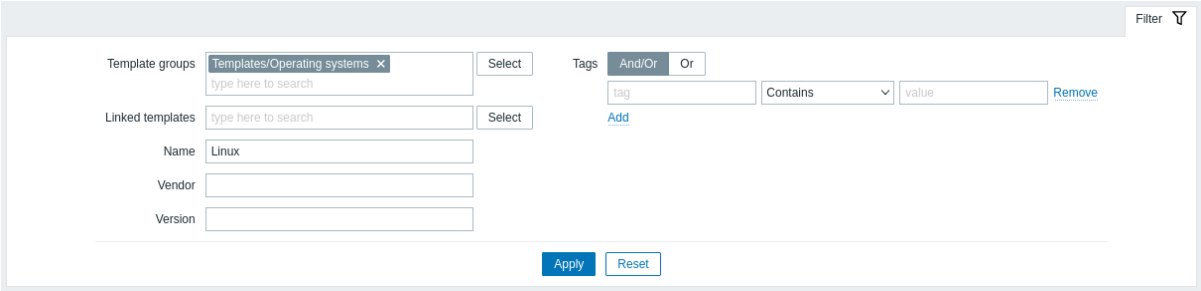
- *Export* - экспортировать шаблон в файл YAML, XML или JSON;
- *Mass update* - обновить несколько свойств для нескольких шаблонов одновременно;
- *Delete* - удалить шаблон, оставив его связанные сущности (элементы данных, триггеры и т. д.) на узлах сети;
- *Delete and clear* - удалить шаблон и его связанные сущности с узлов сети.

Чтобы использовать эти параметры, отметьте флажки перед соответствующими шаблонами, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те шаблоны, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна под кнопками *Create template* и *Import*. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать шаблоны по группе шаблонов, связанным шаблонам, имени и тегам.



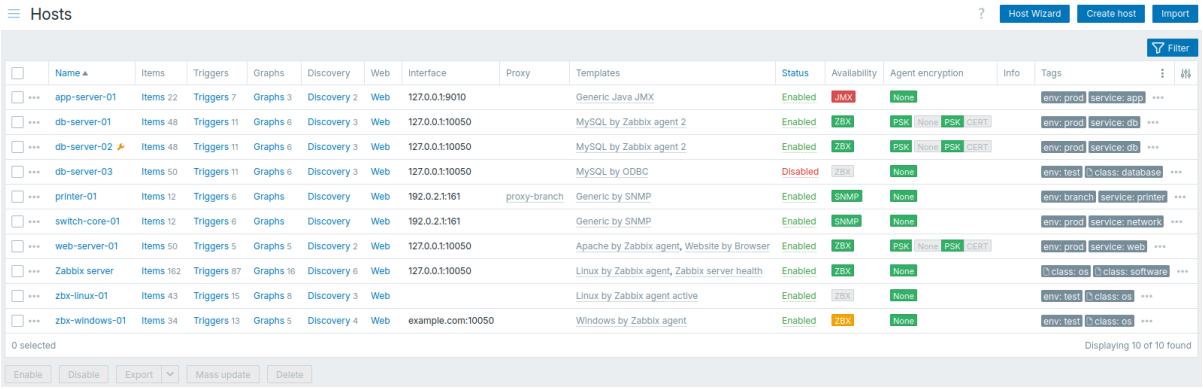
Parameter	Description
Template groups	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы.
Linked templates	Фильтрация по непосредственно связанным шаблонам.
Name	Фильтрация по имени шаблона.
Vendor	Фильтрация по поставщику шаблона.
Version	Фильтрация по версии шаблона.
Tags	Фильтрация по имени и значению тега шаблона. Фильтрация возможна только по тегам уровня шаблона (не унаследованным). Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно выполнения одного условия.

Узлы сети

Обзор

В разделе *Сбор данных* > *Узлы сети* пользователи могут настраивать и обслуживать узлы сети.

Отображается список существующих узлов сети с подробной информацией о них.



Подробная информация об узлах сети отображается в настраиваемой таблице. Кнопка *Настроить таблицу* в заголовке таблицы позволяет отображать, скрывать или изменять порядок столбцов, а также восстанавливать макет по умолчанию. Также можно изменить ширину каждого столбца, перетащив его правую границу; для сброса ширины дважды щелкните границу. Все изменения применяются немедленно и всегда сохраняются для текущего пользователя.

Доступны следующие столбцы:

Столбец	Описание
Имя	Имя узла сети. При нажатии на имя узла сети открывается форма настройки узла сети. Оранжевый значок гаечного ключа после имени узла сети указывает, что этот узел сети находится на обслуживании. Подробная информация об обслуживании отображается при наведении указателя мыши на значок.
Сущности (элементы данных, триггеры, графики, обнаружение, веб)	При нажатии на имя сущности отображаются элементы данных, триггеры и т. д. узла сети. Количество соответствующих сущностей отображается серым цветом.
Интерфейс	Отображается основной интерфейс узла сети.
Прокси	В этом столбце отображаются назначенные прокси: <Имя прокси> - узел сети контролируется автономным прокси (даже если прокси входит в группу прокси); <Имя группы прокси: имя прокси> - узел сети контролируется группой прокси, и сервер Zabbix назначил прокси для контроля узла сети; <Имя группы прокси> - узел сети контролируется группой прокси без прокси либо сервер Zabbix не назначил прокси для контроля узла сети; Ничего - узел сети не контролируется ни прокси, ни группой прокси. Этот столбец отображается только если для параметра фильтра *Контролируется через* установлено значение «Любой», «Прокси» или «Группа прокси».
Шаблоны	Отображаются шаблоны, связанные с узлом сети. Если связанный шаблон содержит другие шаблоны, они отображаются в скобках через запятую. При нажатии на имя шаблона открывается его форма настройки.
Состояние	Отображается состояние узла сети - *Включен* или *Отключен*. Нажав на состояние, можно изменить его вручную. Обнаруженные узлы сети, которые были потеряны, отмечаются информационным значком. Всплывающая подсказка содержит подробную информацию об их состоянии.

Столбец	Описание
Доступность	Отображается доступность узла сети для каждого настроенного интерфейса. Значки доступности показывают текущее состояние интерфейса узла сети на сервере Zabbix. Поэтому при отключении узла сети в веб-интерфейсе его доступность обновится после того, как сервер Zabbix синхронизирует изменения конфигурации. Аналогично, при включении узла сети его доступность обновится после синхронизации изменений конфигурации сервером Zabbix и опроса узла сети. Значки доступности отображаются только для настроенных типов интерфейсов (Agent, SNMP, IPMI, JMX). При наведении указателя мыши на значок открывается всплывающее окно со списком всех интерфейсов того же типа, их подробными сведениями, состоянием и ошибками. Для интерфейса Agent во всплывающем окне отображаются интерфейсы (пассивные) и активные проверки. Если на узле сети настроены только активные проверки, значок интерфейса Agent отображается даже при отсутствии настроенного интерфейса Agent. Для узлов сети без интерфейсов столбец остается пустым. Состояние отдельного интерфейса узла сети определяется соединением между включенным элементом данных, использующим этот интерфейс, и узлом сети. Состояние может быть следующим: **Доступен** - соединение с узлом сети установлено успешно; **Недоступен** - соединение с узлом сети установить не удалось (тайм-аут, проблемы с межсетевым экраном и т. д.); **Неизвестно** - попытка установить соединение с узлом сети не предпринималась либо результат неизвестен. Дополнительную информацию о том, как сервер Zabbix определяет состояние интерфейса, см. в разделах Неизвестное состояние интерфейса и Настройки недоступного/не отвечающего интерфейса узла сети. Состояние всех интерфейсов узла сети одного типа (Agent, SNMP, IPMI, JMX) определяется интерфейсами, которые используются хотя бы одним включенным элементом данных. Состояние обозначается цветом значка: **Зеленый** - все интерфейсы доступны; **Желтый** - как минимум один интерфейс недоступен, а как минимум один доступен или имеет неизвестное состояние; **Красный** - все интерфейсы недоступны; **Серый** - как минимум один интерфейс имеет неизвестное состояние, но недоступных интерфейсов нет. **Доступность активных проверок.** Если на узле сети включена хотя бы одна активная проверка, активные проверки также влияют на общую доступность интерфейса Agent, как описано выше. Для определения доступности активных проверок в потоке активных проверок агента отправляются сообщения проверки связи. Частота отправки сообщений проверки связи задается параметром `HeartbeatFrequency` в конфигурации агента или агента 2 Zabbix (по умолчанию 60 секунд, диапазон 0-3600). Активные проверки считаются недоступными, если с момента сообщения проверки связи прошло более 2 x HeartbeatFrequency секунд. **Примечание:** агенты Zabbix старше версии 6.2.x не отправляют сообщения проверки связи для активных проверок, поэтому доступность их узлов сети остается неизвестной.
Шифрование агента	Отображается состояние шифрования соединений с узлом сети и от узла сети: **Нет** - шифрование не используется; **PSK** - используется предварительно установленный ключ; **Сертификат** - используется сертификат.
Информация	Отображается информация об ошибках (если есть), связанных с узлом сети.
Теги	Отображаются теги узла сети без раскрытия макросов. При наведении указателя мыши на значок с тремя точками ••• или нажатии на него после отображаемых тегов открывается всплывающее окно со всеми тегами. Кнопка ⋮ *Параметры* в заголовке столбца позволяет задать количество отображаемых тегов (1–3), выбрать способ отображения имен тегов (полностью, сокращенно или не отображать), задать приоритет отображения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).

Столбец	Описание
Значение тега	Отображаются значения тегов узла сети без раскрытия макросов. Кнопка ⋮ *Параметры* в заголовке столбца позволяет переименовать столбец, выбрать отображаемые значения тегов или дублировать столбец (дублированные столбцы можно переименовать или удалить).
Пользовательский текст	Отображается пользовательский текст, значения макросов или их сочетание. Кнопка ⋮ *Параметры* в заголовке столбца позволяет переименовать столбец, ввести пользовательский текст или макросы либо дублировать столбец (дублированные столбцы можно переименовать или удалить). Поддерживаемые макросы: `{HOST.*}` (кроме `{HOST.METADATA}`), `{INVENTORY.*}` и пользовательские макросы.

Чтобы открыть меню узла сети, нажмите кнопку ⋮ *Параметры* перед именем узла сети.

Чтобы настроить новый узел сети, нажмите кнопку *Мастер узла сети* или *Создать узел сети* в правом верхнем углу.

Чтобы импортировать узел сети из файла YAML, XML или JSON, нажмите кнопку *Импорт* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько параметров массового редактирования:

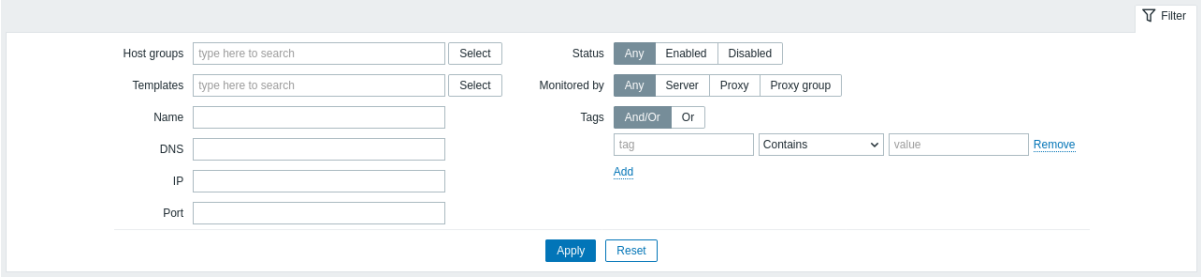
- *Enable* - изменить статус узла сети на *Monitored*;
- *Disable* - изменить статус узла сети на *Not monitored*;
- *Export* - экспортировать узлы сети в файл YAML, XML или JSON;
- *Mass update* - обновить несколько свойств для нескольких узлов сети одновременно;
- *Delete* - удалить узлы сети.

Чтобы использовать эти параметры, отметьте флажки перед соответствующими узлами сети, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас узлы сети. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Фильтр* доступен в правом верхнем углу. При нажатии на него откроется фильтр, где вы можете указать нужные критерии фильтрации.



Parameter	Description
Host groups	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
Templates	Фильтрация по связанным шаблонам.
Name	Фильтрация по видимому имени узла сети.
DNS	Фильтрация по DNS-имени.
IP	Фильтрация по IP-адресу.
Port	Фильтрация по номеру порта.
Status	Фильтрация по статусу узла сети.
Monitored by	Фильтрация узлов сети, которые контролируются сервером Zabbix, прокси или группой прокси.
Proxies	Фильтрация узлов сети, которые контролируются указанными здесь прокси. Это поле доступно только если в поле *Monitored by* выбрано ”Proxy”.
Proxy groups	Фильтрация узлов сети, которые контролируются указанными здесь группами прокси. Это поле доступно только если в поле *Monitored by* выбрано ”Proxy group”.

Parameter	Description
Tags	Фильтрация по имени и значению тега узла сети. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена и значения тегов (с учетом регистра); **Contains** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена и значения тегов (с учетом регистра); **Does not contain** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно выполнения одного условия.

Неизвестный статус интерфейса

Сервер Zabbix отображает статус ”Unknown” для интерфейса узла сети (Agent, SNMP, IPMI, JMX) в следующих случаях:

- Узел сети отключен.
- Узел сети настроен на мониторинг через прокси, другой прокси или сервер, если ранее он мониторился через прокси.
- Узел сети мониторится через прокси, который, по-видимому, находится в автономном режиме (обновления от прокси не получались в течение максимального интервала heartbeat - 1 час).
- Все элементы данных узла сети с этим типом интерфейса отключены.
- Не настроены pollers для этого типа интерфейса (например, параметр конфигурации сервера `StartAgentPollers` или `StartSNMPPollers` установлен в 0).

Доступность интерфейса устанавливается в ”Unknown” после синхронизации кэша конфигурации сервера Zabbix.

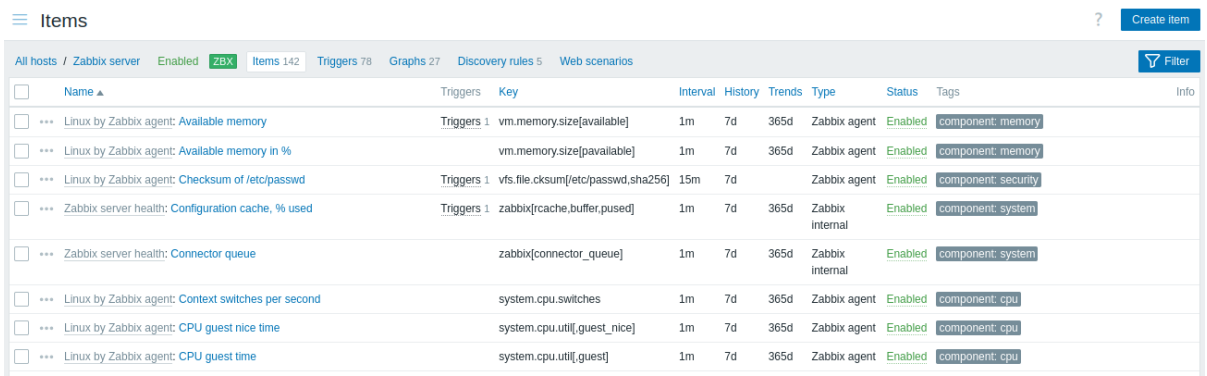
Доступность интерфейса (Available/Not available) на узлах сети, мониторимых через прокси, восстанавливается после синхронизации кэша конфигурации прокси.

Элементы данных

Обзор

Список элементов данных для узла сети можно открыть в разделе *Сбор данных* > *Узлы сети*, нажав *Элементы данных* для соответствующего узла сети.

Отобразится список существующих элементов данных.



Отображаемые данные:

Столбец	Описание
Меню элемента данных	Нажмите значок с тремя точками, чтобы открыть меню элемента данных.
Узел сети	Узел сети, которому принадлежит элемент данных. При нажатии имени узла сети открывается форма настройки узла сети. Этот столбец отображается только в том случае, если в фильтре выбрано несколько узлов сети или ни одного узла сети.
Имя	Имя элемента данных отображается в виде синей ссылки на сведения об элементе данных. При нажатии ссылки с именем элемента данных открывается форма настройки элемента данных. Если элемент данных узла сети принадлежит шаблону, перед его именем отображается имя шаблона в виде серой ссылки. При нажатии ссылки на шаблон открывается список элементов данных на уровне шаблона. Если элемент данных создан на основе прототипа элемента данных, перед его именем оранжевым цветом отображается имя правила обнаружения низкоуровневого обнаружения. При нажатии имени правила обнаружения открывается список прототипов элементов данных.
Триггеры	При наведении указателя мыши на «Триггеры» отображается информационное окно со связанными с элементом данных триггерами. Количество триггеров отображается серым цветом.
Ключ	Отображается ключ элемента данных.
Интервал	Отображается частота проверки. *Примечание*: пассивные элементы данных также можно немедленно проверить, нажав кнопку *Выполнить сейчас* кнопка.
История	Отображается количество дней хранения истории данных элемента данных.
Тренды	Отображается количество дней хранения истории трендов элемента данных.
Тип	Отображается тип элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).
Состояние	Отображается состояние элемента данных: *Включен*, *Отключен* или *Не поддерживается*. Состояние можно изменить вручную, нажав на него: с «Включен» на «Отключен» и обратно; с «Не поддерживается» на «Отключен» и обратно. Обнаруженные элементы данных, которые были потеряны, отмечаются значком информации. Во всплывающей подсказке отображаются сведения об их состоянии.
Теги	Отображаются теги элемента данных. Можно отобразить до трех тегов (пар имя:значение). Если тегов больше, отображается ссылка «...», при наведении на которую можно просмотреть все теги.
Информация	Если элемент данных работает корректно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новый элемент данных, нажмите кнопку *Создать элемент данных* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

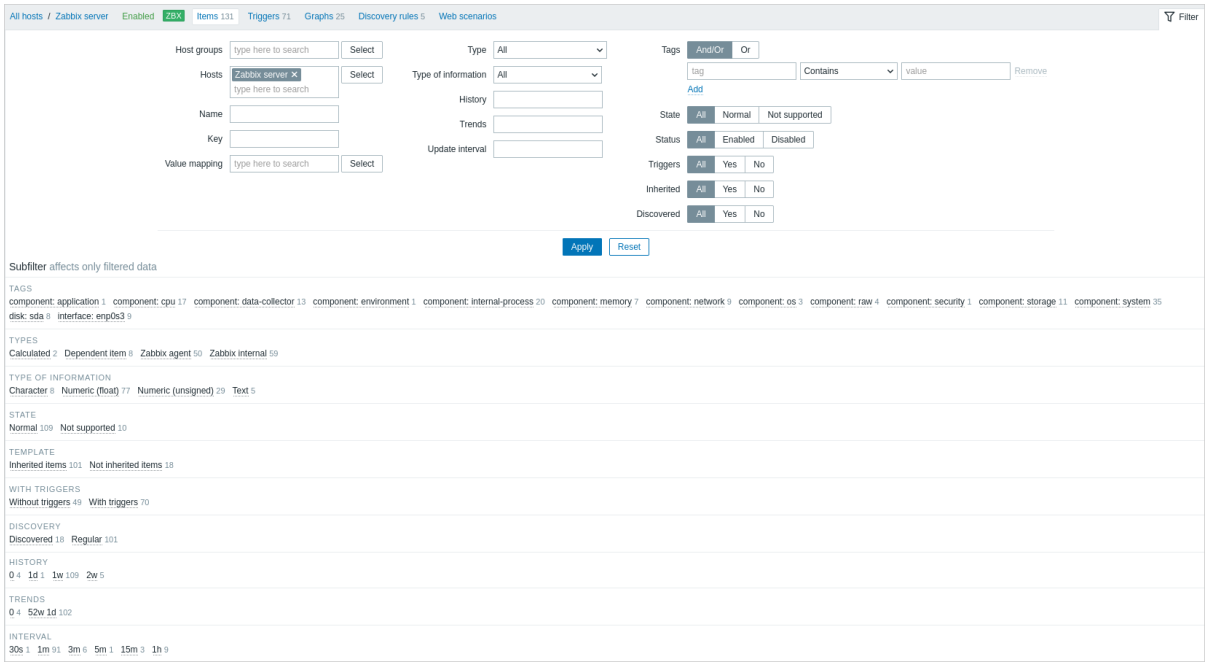
- *Enable* - изменить статус элемента данных на *Enabled*
- *Disable* - изменить статус элемента данных на *Disabled*
- *Execute now* - немедленно выполнить проверку новых значений элемента данных. Поддерживается только для **пассивных** проверок (см. подробнее). Обратите внимание, что при немедленной проверке значений кэш конфигурации не обновляется, поэтому значения не будут отражать самые недавние изменения в конфигурации элемента данных.
- *Clear history and trends* - удалить историю и данные трендов для элементов данных.
- *Copy* - скопировать элементы данных на другие узлы сети или шаблоны.
- *Mass update* - обновить несколько свойств для нескольких элементов данных одновременно.
- *Delete* - удалить элементы данных.

Чтобы использовать эти параметры, установите флажки напротив соответствующих элементов данных, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только интересующие вас элементы данных. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Filter* доступен в правом верхнем углу. При нажатии на него откроется фильтр, где можно указать нужные критерии фильтрации.

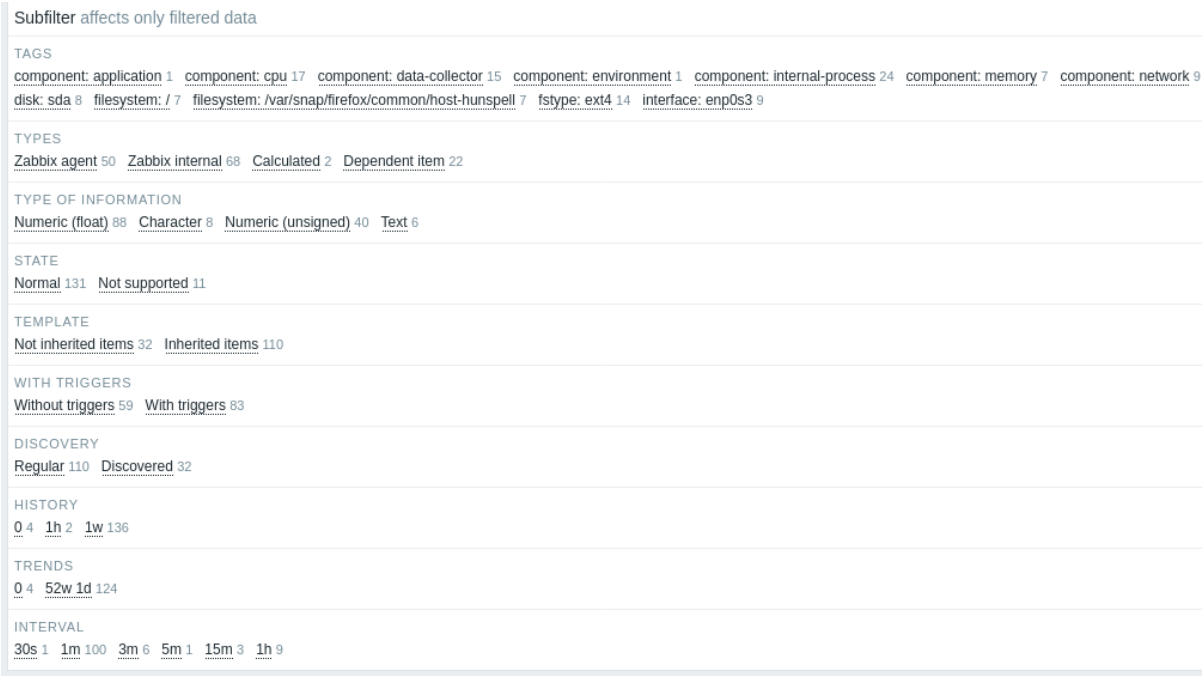


Parameter	Description
Host groups	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Группы узлов сети, содержащие только шаблоны, выбрать нельзя.
Hosts	Фильтрация по одному или нескольким узлам сети.
Name	Фильтрация по имени элемента данных.
Key	Фильтрация по ключу элемента данных.
Value mapping	Фильтрация по используемой карте значений. Этот параметр не отображается, если параметр *Hosts* пуст.
Type	Фильтрация по типу элемента данных (Zabbix agent, SNMP agent и т. д.).
Type of information	Фильтрация по типу информации (беззнаковое число, число с плавающей точкой и т. д.).
History	Фильтрация по времени хранения истории элемента данных.
Trends	Фильтрация по времени хранения трендов элемента данных.
Update interval	Фильтрация по интервалу обновления элемента данных.
Tags	Укажите теги, чтобы ограничить количество отображаемых элементов данных. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия
State	Фильтрация по состоянию элемента данных - *Normal* или *Not supported*.
Status	Фильтрация по статусу элемента данных - *Enabled* или *Disabled*.
Triggers	Фильтрация элементов данных с триггерами или без них.
Inherited	Фильтрация элементов данных, унаследованных или не унаследованных от шаблона.
Discovery	Фильтрация элементов данных, обнаруженных или не обнаруженных низкоуровневым обнаружением.

Использование подфильтра

Подфильтр позволяет дополнительно изменить фильтрацию, примененную основным фильтром.

Он содержит ссылки, по которым можно щелкнуть для быстрого доступа к связанным элементам данных. Элементы данных связаны общей сущностью: тегом, типом элемента данных, состоянием элемента данных, интервалом обновления элемента данных и т. д. При выборе сущности она выделяется серым фоном, а элементы данных немедленно фильтруются (нет необходимости нажимать *Применить* в основном фильтре). Выбор другой сущности добавляет ее в результаты фильтрации. Повторный выбор сущности отменяет фильтрацию по ней.



Подфильтры создаются на основе отфильтрованных данных, количество которых ограничено 1000 записями. Чтобы отображать в подфильтре больше записей, увеличьте значение параметра *Ограничение результатов поиска и фильтрации* (раздел *Администрирование* > *Общие* > *Веб-интерфейс*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы информация о доступных параметрах фильтрации и их количестве всегда оставалась актуальной.

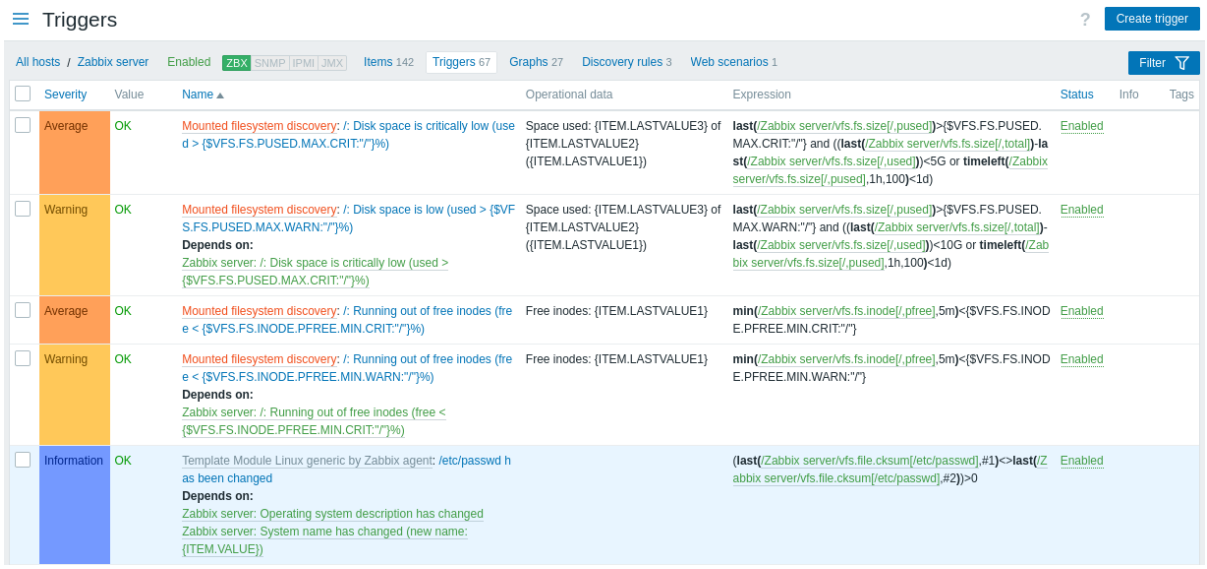
Количество отображаемых сущностей ограничено 100 по горизонтали. Если сущностей больше, в конце отображается значок с тремя точками; выбрать его нельзя.

Число рядом с каждой доступной для выбора сущностью указывает количество сгруппированных в ней элементов данных (на основе результатов основного фильтра). После выбора сущности рядом с другими доступными сущностями отображаются числа со знаком «плюс», указывающие, сколько элементов данных можно добавить к текущему выбору. Сущности, не содержащие элементов данных, не отображаются, если только они ранее не были выбраны в подфильтре.

Триггеры

Обзор

Список триггеров для узла сети можно открыть в разделе *Сбор данных* > *Узлы сети*, нажав *Триггеры* для соответствующего узла сети.



Отображаемые данные:

Столбец	Описание
Важность	Важность триггера отображается как его названием, так и цветом фона ячейки.
Значение	Отображается состояние триггера: **OK** - состояние «OK» **PROBLEM** - состояние «Проблема»
Узел сети	Узел сети, к которому относится триггер. При нажатии на имя узла сети открывается форма настройки узла сети. Этот столбец отображается только в том случае, если в фильтре выбрано несколько узлов сети или ни одного узла сети.
Имя	Имя триггера отображается в виде синей ссылки на сведения о триггере. При нажатии на ссылку с именем триггера открывается форма настройки триггера. Если триггер узла сети принадлежит шаблону, перед именем триггера отображается имя шаблона в виде серой ссылки. При нажатии на ссылку с именем шаблона открывается список триггеров на уровне шаблона. Если триггер создан на основе прототипа триггера, перед его именем оранжевым цветом отображается имя правила обнаружения низкоуровневого обнаружения. При нажатии на имя правила обнаружения открывается список прототипов триггеров.
Операционные данные	Определение операционных данных триггера, содержащее произвольные строки и макросы, значения которых динамически подставляются в разделе *Мониторинг* > *Проблемы*.
Выражение	Отображается выражение триггера. Часть выражения, содержащая узел сети и элемент данных, отображается в виде ссылки, ведущей к форме настройки элемента данных.
Состояние	Отображается состояние триггера: *Включен*, *Отключен* или *Неизвестно*. Нажав на состояние, его можно изменить вручную: с «Включен» на «Отключен» и обратно; с «Неизвестно» на «Отключен» и обратно. Проблемы отключенного триггера больше не отображаются в веб-интерфейсе, но не удаляются. Обнаруженные триггеры, которые были потеряны, помечаются значком информации. Во всплывающей подсказке отображаются сведения об их состоянии.
Информация	Если все работает правильно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.
Теги	Если триггер содержит теги, в этом столбце отображаются имя и значение тега.

Чтобы настроить новый триггер, нажмите кнопку *Создать триггер* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Enable* - изменить статус триггера на *Enabled*.
- *Disable* - изменить статус триггера на *Disabled*.
- *Copy* - скопировать триггеры на другие узлы сети или шаблоны.

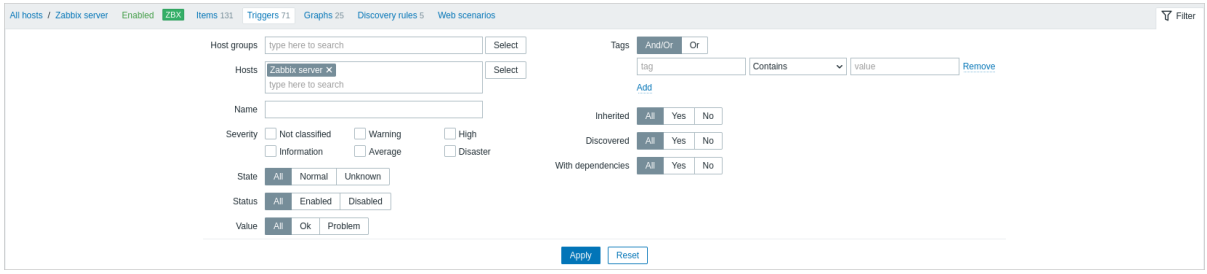
- *Mass update* - обновить несколько свойств для ряда триггеров одновременно.
- *Delete* - удалить триггеры.

Чтобы использовать эти параметры, отметьте флажки перед соответствующими триггерами, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те триггеры, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Значок *Filter* доступен в правом верхнем углу. При нажатии на него откроется фильтр, в котором можно указать нужные критерии фильтрации.



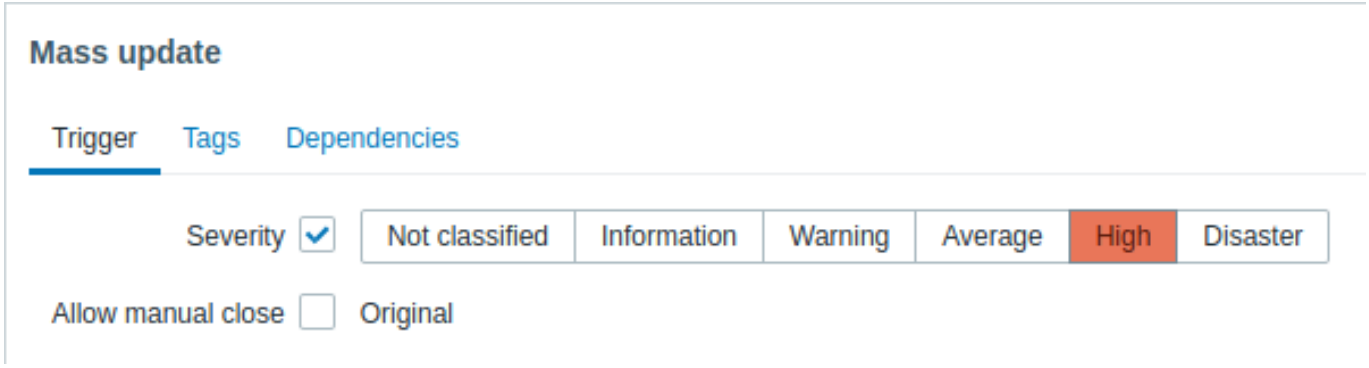
Parameter	Description
Host groups	Фильтрация по одной или нескольким группам узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети. Группы узлов сети, содержащие только шаблоны, выбрать нельзя.
Hosts	Фильтрация по одному или нескольким узлам сети. Если выше уже выбраны группы узлов сети, выбор узлов сети ограничивается этими группами.
Name	Фильтрация по имени триггера.
Severity	Выберите, чтобы фильтровать по одной или нескольким степеням важности триггера.
State	Фильтрация по состоянию выражения триггера (Normal/Unknown).
Status	Фильтрация по статусу активации триггера (Enabled/Disabled).
Value	Фильтрация по значению триггера.
Tags	Фильтрация по имени и значению тега триггера. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов **Equals** - включить указанные имена тегов и значения (с учетом регистра) **Contains** - включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Does not exist** - исключить указанные имена тегов **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра) **Does not contain** - исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра) Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Or **Or** - достаточно выполнения одного условия Макросы и функции макросов поддерживаются как в поле имени тега, так и в поле значения тега.
Inherited	Фильтрация триггеров, унаследованных (или не унаследованных) от шаблона.
Discovered	Фильтрация триггеров, обнаруженных (или не обнаруженных) низкоуровневым обнаружением.
With dependencies	Фильтрация триггеров с зависимостями (или без зависимостей).

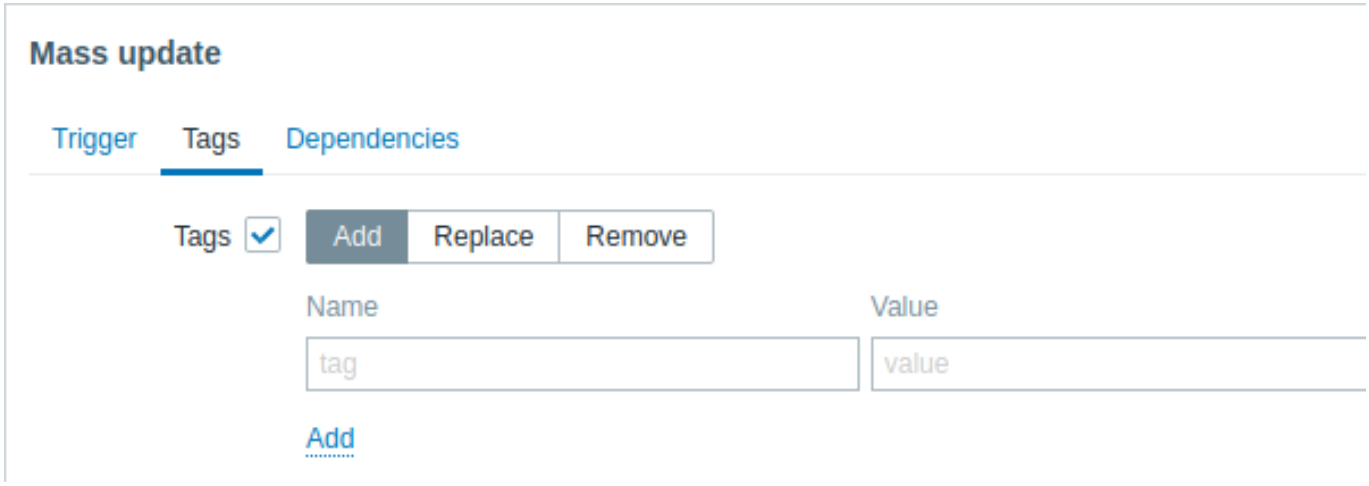
Массовое обновление

С помощью массового обновления можно одновременно изменить некоторые атрибуты нескольких триггеров, избавив себя от необходимости открывать каждый отдельный триггер для редактирования.

Чтобы выполнить массовое обновление триггеров, выполните следующие действия:

- Установите флажки напротив триггеров, которые необходимо обновить, в списке.
- Нажмите *Массовое обновление* под списком.
- Перейдите на вкладку с необходимыми атрибутами (*Триггер*, *Теги* или *Зависимости*).
- Установите флажки напротив атрибутов, которые необходимо обновить.

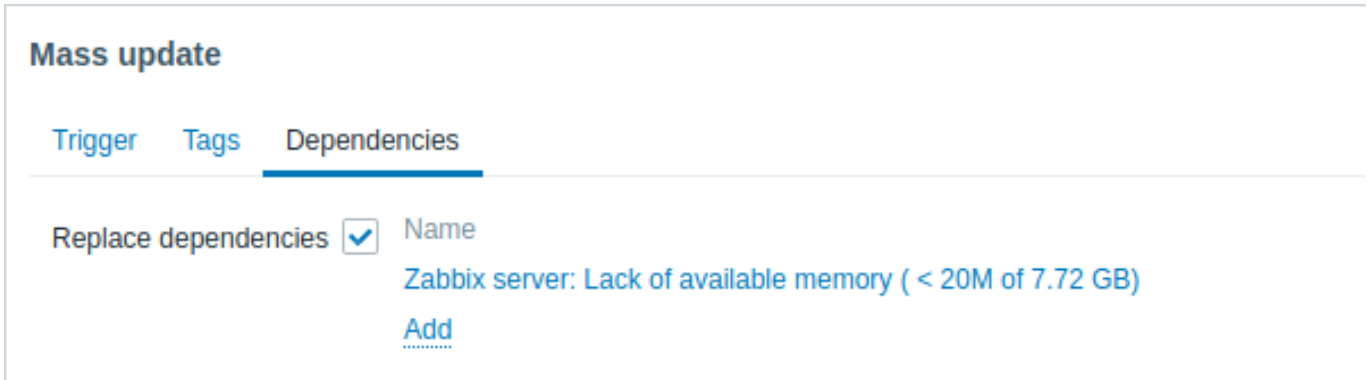




При выборе соответствующей кнопки для обновления тегов доступны следующие варианты:

- *Добавить* - добавить новые теги для триггеров;
- *Заменить* - удалить все существующие теги триггера и заменить их указанными ниже тегами;
- *Удалить* - удалить указанные теги из триггеров.

Обратите внимание, что теги с одинаковым именем, но разными значениями, не считаются «дубликатами» и могут быть добавлены к одному и тому же триггеру.



Заменить зависимости - удалить все существующие зависимости триггера и заменить их указанными зависимостями.

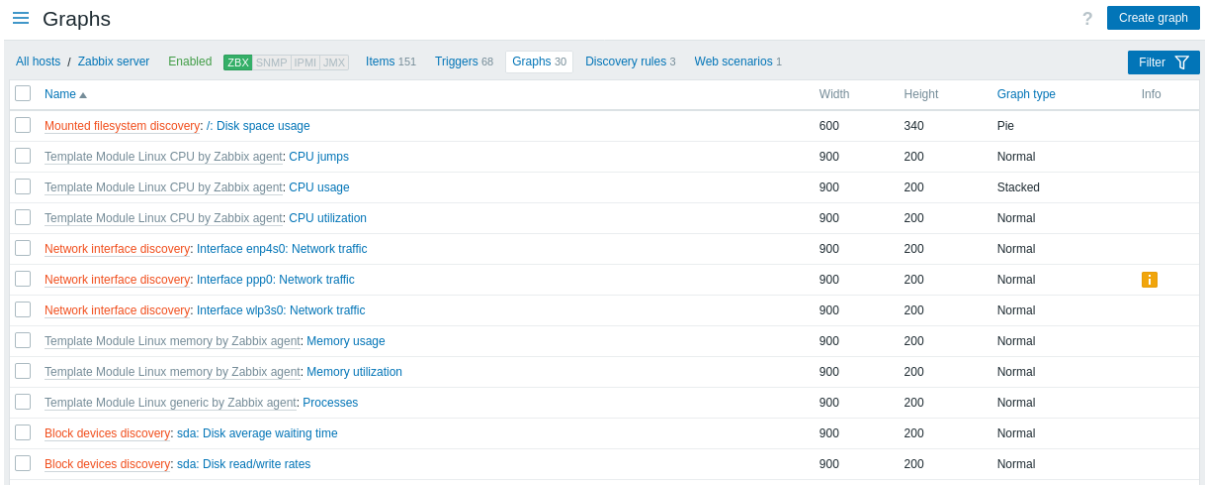
Нажмите *Обновить*, чтобы применить изменения.

Графики

Обзор

Список пользовательских графиков для узла сети можно открыть в разделе *Сбор данных* > *Узлы сети*, нажав *Графики* для соответствующего узла сети.

Отображается список существующих графиков.



Отображаемые данные:

Столбец	Описание
Узел сети	Узел сети, к которому относится график. При нажатии на имя узла сети открывается форма настройки узла сети. Этот столбец отображается только в том случае, если в фильтре выбрано несколько узлов сети или ни одного узла сети.
Имя	Имя пользовательского графика, отображаемое в виде синей ссылки на сведения о графике. При нажатии на ссылку с именем графика открывается форма настройки графика. Если график узла сети принадлежит шаблону, перед именем графика отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список графиков на уровне шаблона. Если график создан на основе прототипа графика, перед его именем оранжевым цветом отображается имя правила обнаружения низкоуровневого обнаружения. При нажатии на имя правила обнаружения открывается список прототипов графиков.
Ширина	Отображается ширина графика.
Высота	Отображается высота графика.
Тип графика	Отображается тип графика: *Обычный*, *С накоплением*, *Круговая диаграмма* или *Взрывная диаграмма*.
Информация	Если график работает корректно, в этом столбце ничего не отображается. В случае ошибок отображается квадратный значок с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новый график, нажмите кнопку *Создать график* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Копировать* - копировать графики на другие узлы сети или шаблоны
- *Удалить* - удалить графики

Чтобы использовать эти параметры, отметьте флажки перед соответствующими графиками, затем нажмите нужную кнопку.

Использование фильтра

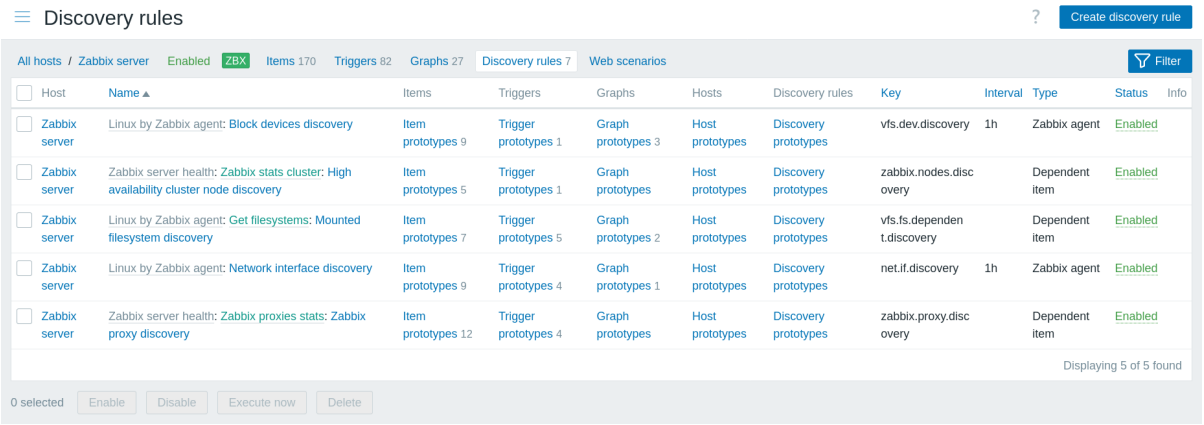
Вы можете фильтровать графики по группе узлов сети и узлу сети. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Правила обнаружения

Обзор

Список правил обнаружения низкого уровня для узла сети можно открыть в разделе *Сбор данных* > *Узлы сети*, нажав *Обнаружение* для соответствующего узла сети.

Отображается список существующих правил обнаружения низкого уровня. Также можно просмотреть все правила обнаружения независимо от узла сети или все правила обнаружения определенной группы узлов сети, изменив настройки фильтра.



Отображаемые данные:

Столбец	Описание
Узел сети	Отображается видимое имя узла сети. При нажатии на имя узла сети открывается форма настройки узла сети. Если видимое имя узла сети отсутствует, отображается техническое имя узла сети.
Имя	Имя правила отображается в виде синей ссылки. При нажатии на имя правила открывается форма настройки правила обнаружения низкого уровня. Если правило обнаружения принадлежит шаблону, перед именем правила отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список правил на уровне шаблона.
Элементы данных	Отображается ссылка на список прототипов элементов данных. Количество существующих прототипов элементов данных отображается серым цветом.
Триггеры	Отображается ссылка на список прототипов триггеров. Количество существующих прототипов триггеров отображается серым цветом.
Графики	Отображается ссылка на список прототипов графиков. Количество существующих прототипов графиков отображается серым цветом.
Узлы сети	Отображается ссылка на список прототипов узлов сети. Количество существующих прототипов узлов сети отображается серым цветом.
Ключ	Отображается ключ элемента данных, используемый для обнаружения.
Интервал	Отображается периодичность выполнения обнаружения. *Примечание*: обнаружение также можно выполнить немедленно, нажав кнопку *Выполнить сейчас* под списком.
Тип	Отображается тип элемента данных, используемый для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Состояние	Отображается состояние правила обнаружения: *Включено*, *Отключено* или *Не поддерживается*. Нажав на состояние, его можно изменить: с «Включено» на «Отключено» и обратно; с «Не поддерживается» на «Отключено» и обратно.
Информация	Если все в порядке, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новое правило обнаружения низкого уровня, нажмите кнопку *Создать правило обнаружения* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

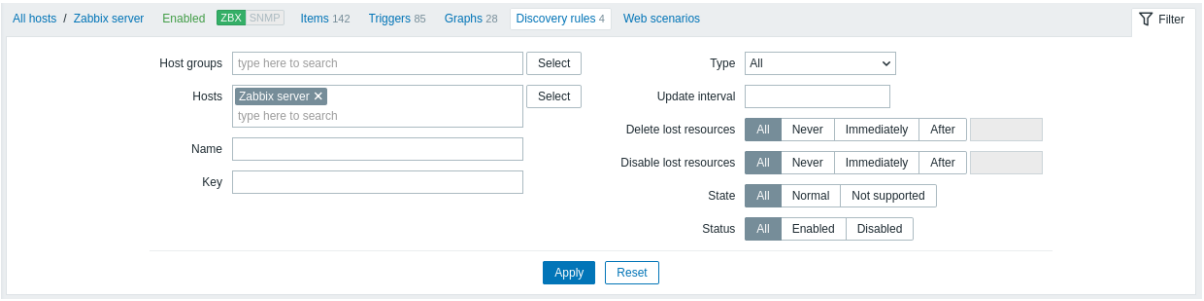
- *Enable* - изменить статус правила низкоуровневого обнаружения на *Enabled*.
- *Disable* - изменить статус правила низкоуровневого обнаружения на *Disabled*.
- *Execute now* - немедленно выполнить обнаружение на основе правил обнаружения. См. подробнее. Обратите внимание, что при немедленном выполнении обнаружения кэш конфигурации не обновляется, поэтому результат не будет отражать самые последние изменения в конфигурации правила обнаружения.
- *Delete* - удалить правила низкоуровневого обнаружения.

Чтобы использовать эти параметры, отметьте флажки перед нужными правилами обнаружения, затем нажмите требуемую кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила обнаружения, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком правил обнаружения. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать правила обнаружения по группе узлов сети, узлу сети, имени, ключу элемента данных, типу элемента данных и другим параметрам.

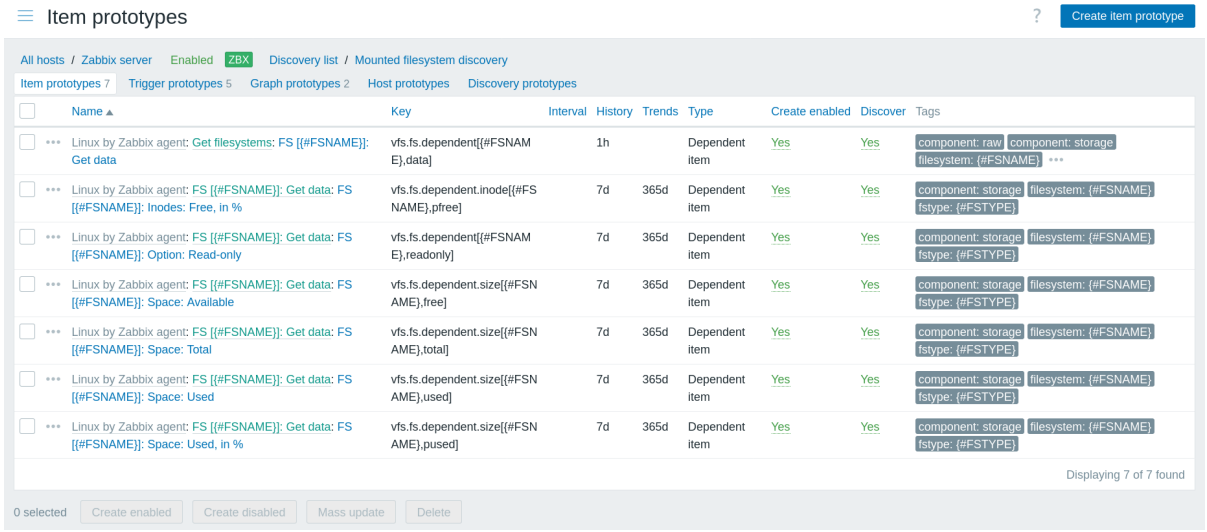


Parameter	Description
Host groups	Фильтр по одной или нескольким группам узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети.
Hosts	Фильтр по одному или нескольким узлам сети.
Name	Фильтр по имени правила обнаружения.
Key	Фильтр по ключу элемента данных обнаружения.
Type	Фильтр по типу элемента данных обнаружения.
Update interval	Фильтр по интервалу обновления. Недоступно для Zabbix trapper и зависимых элементов данных.
Delete lost resources	Фильтр по периоду *Delete lost resources*.
Disable lost resources	Фильтр по периоду *Disable lost resources*.
SNMP OID	Фильтр по SNMP OID. Доступно только если в качестве типа выбран *SNMP agent*.
State	Фильтр по состоянию правила обнаружения (All/Normal/Not supported).
Status	Фильтр по статусу правила обнаружения (All/Enabled/Disabled).

Прототипы элементов данных

Обзор

В этом разделе отображаются прототипы элементов данных правила обнаружения низкоуровневых объектов на узле сети. Прототипы элементов данных являются основой реальных элементов данных узла сети, которые создаются во время низкоуровневого обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа элемента данных отображается в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа элемента данных. Если прототип элемента данных принадлежит шаблону, перед именем правила отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов элементов данных на уровне шаблона. Если прототип элемента данных имеет тип «Зависимый элемент данных», перед именем прототипа элемента данных зеленым цветом отображается имя главного элемента данных.
Ключ	Отображается ключ прототипа элемента данных.
Интервал	Отображается частота проверки.
История	Отображается количество дней хранения истории данных элемента данных.
Тренды	Отображается количество дней хранения истории трендов элемента данных.
Тип	Отображается тип прототипа элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).
Создание включено	Создание элемента данных на основе этого прототипа: **Да** - включено **Нет** - отключено. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Обнаружение	Обнаружение элемента данных на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Теги	Отображаются теги прототипа элемента данных.

Чтобы настроить новый прототип элемента данных, нажмите кнопку *Создать прототип элемента данных* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

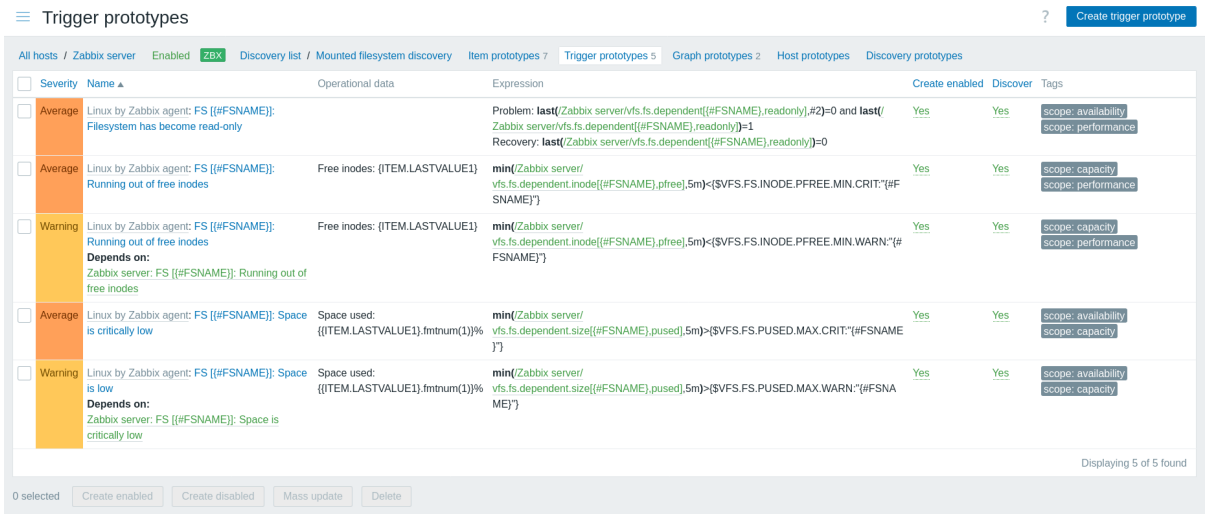
- *Создать включенными* - создать эти элементы данных как *Включенные*
- *Создать отключенными* - создать эти элементы данных как *Отключенные*
- *Массовое обновление* - массово обновить эти прототипы элементов данных
- *Удалить* - удалить эти прототипы элементов данных

Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами элементов данных, затем нажмите нужную кнопку.

Прототипы триггеров

Обзор

В этом разделе отображаются прототипы триггеров правила обнаружения низкого уровня на узле сети. Прототипы триггеров являются основой реальных триггеров узла сети, которые создаются во время обнаружения низкого уровня.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа триггера, отображаемое в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа триггера. Если прототип триггера принадлежит связанному шаблону, перед именем триггера отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов триггеров на уровне связанного шаблона.
Операционные данные	Отображается формат операционных данных триггера, содержащий произвольные строки и макросы, значения которых динамически подставляются в разделе *Мониторинг > Проблемы*.
Создание включено	Создание триггера на основе этого прототипа: **Да** - включено **Нет** - отключено. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Обнаружение	Обнаружение триггера на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Теги	Отображаются теги прототипа триггера.

Чтобы настроить новый прототип триггера, нажмите кнопку *Создать прототип триггера* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

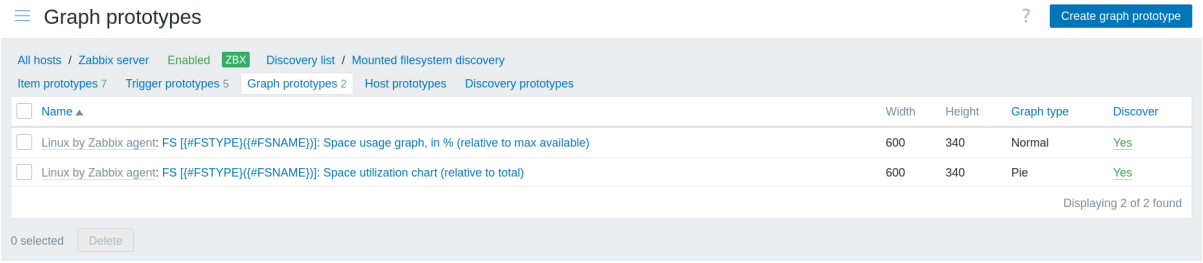
- *Создать включенными* - создать эти триггеры как *Включенные*
- *Создать отключенными* - создать эти триггеры как *Отключенные*
- *Массовое обновление* - массово обновить эти прототипы триггеров
- *Удалить* - удалить эти прототипы триггеров

Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами триггеров, затем нажмите нужную кнопку.

Прототипы графиков

Обзор

В этом разделе отображаются прототипы графиков правила обнаружения низкого уровня на узле сети. Прототипы графиков являются основой реальных графиков узла сети, которые создаются в процессе обнаружения низкого уровня.



Отображаемые данные:

Column	Description
Name	Имя прототипа графика отображается в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа графика. Если прототип графика принадлежит связанному шаблону, перед именем графика отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов графиков на уровне связанного шаблона.
Width	Отображается ширина прототипа графика.
Height	Отображается высота прототипа графика.
Type	Отображается тип прототипа графика: *Normal*, *Stacked*, *Pie* или *Exploded*.
Discover	Обнаружение графика на основе этого прототипа: **Yes** - обнаруживать **No** - не обнаруживать. Переключаться между значениями «Yes» и «No» можно, нажимая на них.

Чтобы настроить новый прототип графика, нажмите кнопку *Create graph prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

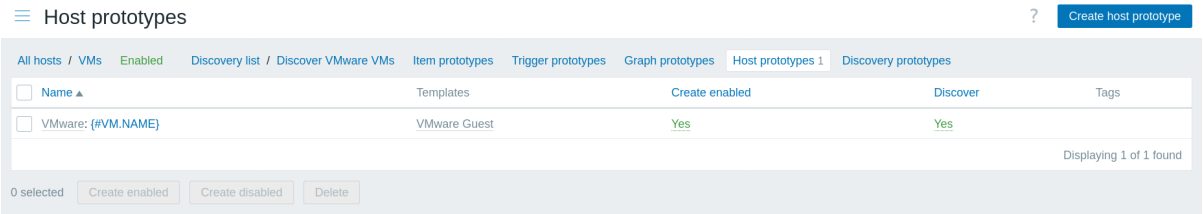
- *Удалить* - удалить эти прототипы графиков

Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами графиков, затем нажмите нужную кнопку.

Прототипы узлов сети

Обзор

В этом разделе отображаются прототипы узлов сети правила обнаружения низкого уровня на узле сети. Прототипы узлов сети являются основой реальных узлов сети, создаваемых в процессе обнаружения низкого уровня.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа узла сети, отображаемое в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа узла сети. Если прототип узла сети принадлежит связанному шаблону, перед именем узла сети отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов узлов сети на уровне связанного шаблона.
Шаблоны	Отображаются шаблоны прототипа узла сети.
Создание включено	Создавать узел сети на основе этого прототипа: **Да** - включено **Нет** - отключено. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.

Столбец	Описание
Обнаружение	Обнаруживать узел сети на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Переключаться между значениями «Да» и «Нет» можно, нажимая на них.
Теги	Отображаются теги прототипа узла сети.

Чтобы настроить новый прототип узла сети, нажмите кнопку *Создать прототип узла сети* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

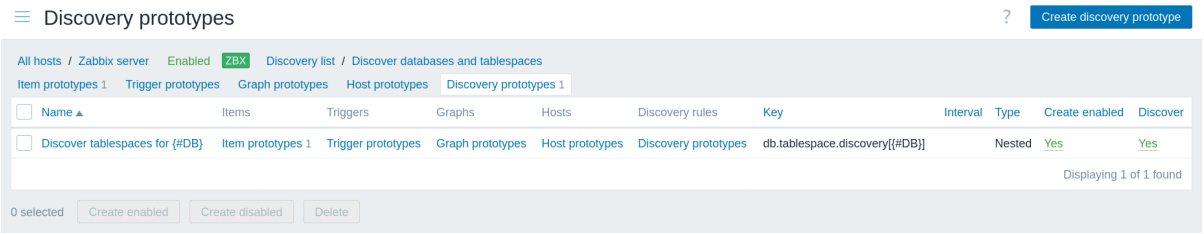
- *Создать включенными* - создать эти узлы сети как *Включенные*
- *Создать отключенными* - создать эти узлы сети как *Отключенные*
- *Удалить* - удалить эти прототипы узлов сети

Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами узлов сети, затем нажмите нужную кнопку.

Прототипы обнаружения

Обзор

В этом разделе отображаются прототипы обнаружения правила обнаружения низкоуровневого обнаружения на узле сети. Прототипы обнаружения являются вложенными правилами обнаружения родительского правила обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа обнаружения отображается в виде синей ссылки. При нажатии на имя открывается форма настройки прототипа обнаружения. Если прототип обнаружения принадлежит связанному шаблону, перед именем узла сети отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список прототипов обнаружения на уровне связанного шаблона.
Элементы данных	Отображается ссылка на список прототипов элементов данных. Количество существующих прототипов элементов данных отображается серым цветом.
Триггеры	Отображается ссылка на список прототипов триггеров. Количество существующих прототипов триггеров отображается серым цветом.
Графики	Отображается ссылка на список прототипов графиков. Количество существующих прототипов графиков отображается серым цветом.
Узлы сети	Отображается ссылка на список прототипов узлов сети. Количество существующих прототипов узлов сети отображается серым цветом.
Правила обнаружения	Отображается ссылка на список прототипов обнаружения. Количество существующих прототипов обнаружения отображается серым цветом.
Ключ	Отображается ключ элемента данных, используемый для обнаружения.
Интервал	Отображается частота выполнения обнаружения. *Примечание*: обнаружение также можно выполнить немедленно, нажав кнопку *Выполнить сейчас* под списком.
Тип	Отображается тип элемента данных, используемый для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Создание включено	Отображается статус создания прототипа обнаружения: создание включено (*Да*) или создание отключено (*Нет*). Нажав на статус, его можно изменить с «Да» на «Нет» и наоборот.

Столбец	Описание
Обнаружение	Отображается статус обнаружения прототипа обнаружения: обнаруживать (*Да*) или не обнаруживать (*Нет*). Нажав на статус, его можно изменить с «Да» на «Нет» и наоборот.

Чтобы настроить новый прототип обнаружения, нажмите кнопку *Создать прототип обнаружения* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Создать включенными* - создать эти прототипы обнаружения как *Включенные*
- *Создать отключенными* - создать эти прототипы обнаружения как *Отключенные*
- *Удалить* - удалить эти прототипы обнаружения

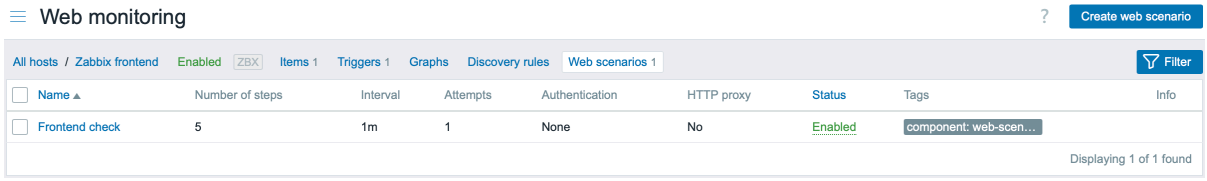
Чтобы использовать эти параметры, отметьте флажки перед соответствующими прототипами обнаружения, затем нажмите нужную кнопку.

Веб-сценарии

Обзор

Список веб-сценариев для узла сети можно открыть в разделе *Сбор данных* > *Узлы сети*, нажав *Веб* для соответствующего узла сети.

Отображается список существующих веб-сценариев.



Отображаемые данные:

Столбец	Описание
Имя	Имя веб-сценария. При нажатии на имя веб-сценария открывается форма настройки веб-сценария. Если веб-сценарий узла сети принадлежит шаблону, перед именем веб-сценария отображается имя шаблона в виде серой ссылки. При нажатии на ссылку шаблона открывается список веб-сценариев на уровне шаблона.
Количество шагов	Количество шагов в сценарии.
Интервал обновления	Частота выполнения сценария.
Попытки	Количество попыток выполнения шагов веб-сценария.
Аутентификация	Отображается метод аутентификации: Basic, NTLM, Kerberos, Digest или None.
HTTP-прокси	Отображается HTTP-прокси или «Нет», если он не используется.
Состояние	Отображается состояние веб-сценария: *Включен* или *Отключен*. Нажав на состояние, его можно изменить.
Теги	Отображаются теги веб-сценария. Можно отобразить до трех тегов (пар «имя:значение»). Если тегов больше, отображается ссылка «...», при наведении на которую можно просмотреть все теги.
Информация	Если все работает корректно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новый веб-сценарий, нажмите кнопку *Создать веб-сценарий* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Включить* - изменить статус сценария на *Включен*
- *Отключить* - изменить статус сценария на *Отключен*
- *Очистить историю и тренды* - очистить данные истории и трендов для сценариев

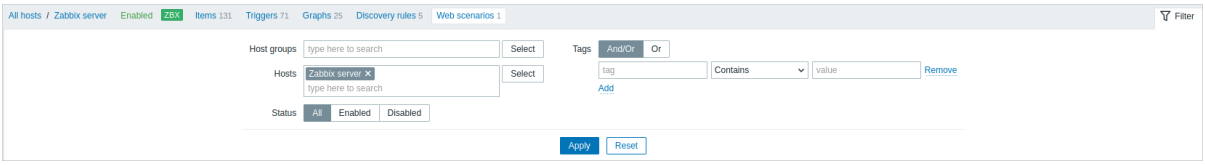
- *Удалить* - удалить веб-сценарии

Чтобы использовать эти параметры, отметьте флажки перед соответствующими веб-сценариями, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те сценарии, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком веб-сценариев. При нажатии на нее откроется фильтр, где можно указать нужные критерии фильтрации.



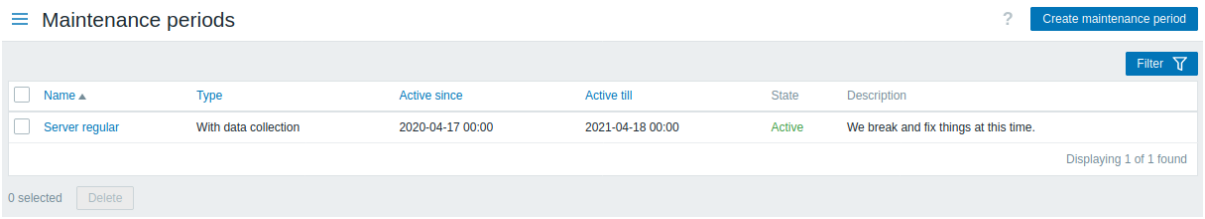
Parameter	Description
Host groups	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
Hosts	Фильтрация по одному или нескольким узлам сети. Если выше уже выбраны группы узлов сети, выбор узлов сети ограничивается этими группами.
Status	Фильтрация по статусу веб-сценария.
Tags	Фильтрация по имени и значению тега веб-сценария. Можно включать, а также исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны следующие операторы: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена тегов и значения (с учетом регистра); **Contains** - включить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена тегов и значения (с учетом регистра); **Does not contain** - исключить указанные имена тегов, значения которых содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **And/Or** - должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Or; **Or** - достаточно, чтобы было выполнено одно условие.

Обслуживание

Обзор

В разделе *Сбор данных* > *Обслуживание* пользователи могут настраивать и управлять периодами обслуживания для узлов сети.

Отображается список существующих периодов обслуживания с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя периода обслуживания. При нажатии на имя периода обслуживания открывается форма настройки периода обслуживания.
Тип	Отображается тип обслуживания: *Сбор данных* или *Без сбора данных*
Активен с	Дата и время, начиная с которых выполнение периодов обслуживания становится активным. Примечание: это время не активирует период обслуживания; периоды обслуживания необходимо задавать отдельно.
Активен до	Дата и время, до которых выполнение периодов обслуживания остается активным.
Состояние	Состояние периода обслуживания: **Предстоящий** - скоро станет активным **Активный** - активен **Истекший** - больше не активен
Описание	Отображается описание периода обслуживания.

Чтобы настроить новый период обслуживания, нажмите кнопку *Создать период обслуживания* в правом верхнем углу.

Параметры массового редактирования

Кнопка под списком предлагает один вариант массового редактирования:

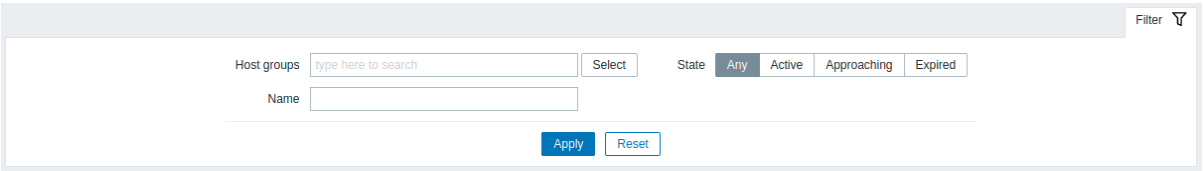
- *Удалить* - удалить периоды обслуживания

Чтобы использовать этот вариант, отметьте флажки перед соответствующими периодами обслуживания и нажмите *Удалить*.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те периоды обслуживания, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком периодов обслуживания. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать периоды обслуживания по группе узлов сети, имени и состоянию.



Расчет очередей во время обслуживания

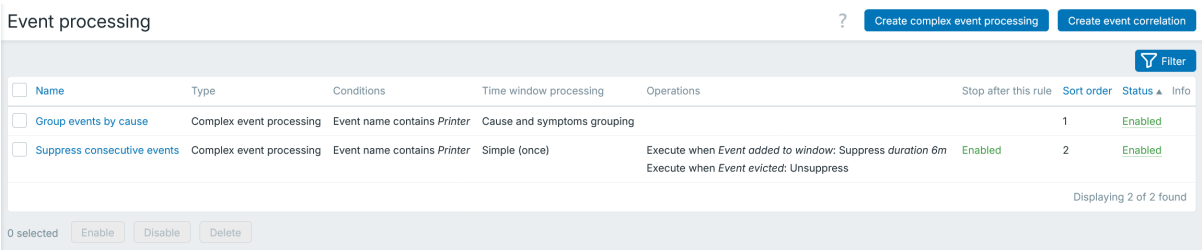
Attention:
Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Расчет очередей во время обслуживания.

Event processing

Overview

In the *Data collection* > *Event processing* section users can configure and maintain:

- Complex event processing rules (CEP rules)
- Global event correlation rules (deprecated)



Displayed data:

Column	Description
Name	Name of the rule. Clicking the name opens the rule configuration form - for CEP rule or global correlation rule respectively.
Type	Type of the rule - *Complex event processing* or *Correlation*.
Conditions	Rule conditions are displayed.
Time window processing	Time window type is displayed.
Operations	Rule operations are displayed.
Stop after this rule	Stop execution after this rule - *Enabled* or *Disabled*.
Sort order	Sort order of the CEP rule.
Status	Rule status is displayed - *Enabled* or *Disabled*. By clicking the status you can change it.
Info	If everything is working correctly, no icon is displayed in this column. In case of errors, a square icon with the letter "i" is displayed. Hover over the icon to see a tooltip with the error description.

To configure a new CEP rule, click the *Create complex event processing* button in the upper-right corner.

To configure a new correlation rule, click the *Create event correlation* button in the upper-right corner.

Mass editing options

Buttons below the list offer some mass-editing options:

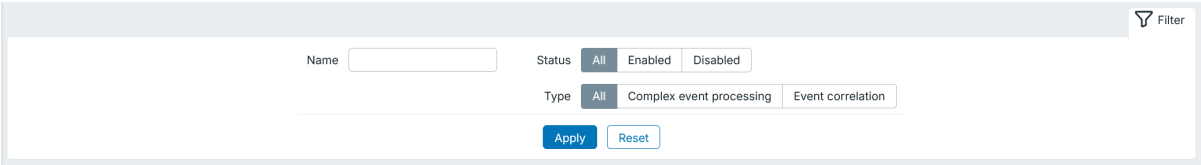
- *Enable* - change the rule status to *Enabled*
- *Disable* - change the rule status to *Disabled*
- *Delete* - delete the rules

To use these options, mark the checkboxes before the respective rules, then click the required button.

Using filter

You can use the filter to display only the rules you are interested in. For better search performance, data is searched with macros unresolved.

The *Filter* link is available above the list of CEP/correlation rules. If you click it, a filter becomes available where you can filter rules by name and status.

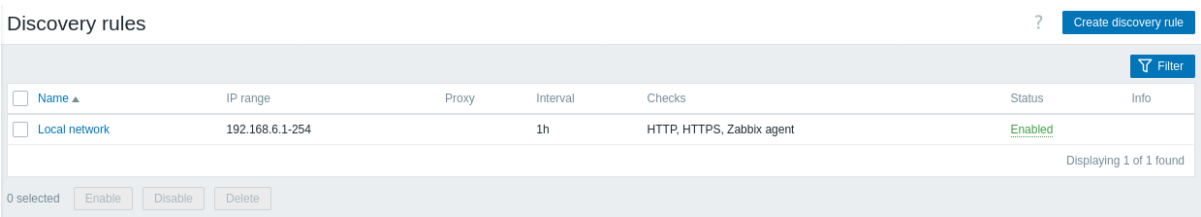


Обнаружение

Обзор

В разделе *Сбор данных* > *Обнаружение* пользователи могут настраивать и обслуживать правила обнаружения.

Отображается список существующих правил обнаружения с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя правила обнаружения. При нажатии на имя правила обнаружения открывается форма настройки правила обнаружения.
Диапазон IP-адресов	Отображается диапазон IP-адресов, используемый для сканирования сети.
Прокси	Отображается имя прокси, если обнаружение выполняется прокси.
Интервал	Отображается периодичность выполнения обнаружения.

Столбец	Описание
Проверки	Отображаются типы проверок, используемых для обнаружения.
Состояние	Отображается состояние правила обнаружения: *Включено* или *Отключено*. Нажав на состояние, его можно изменить.
Информация	Если все работает корректно, этот столбец остается пустым. В случае ошибок отображается красный значок информации с буквой «i». Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новое правило обнаружения, нажмите кнопку *Создать правило обнаружения* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

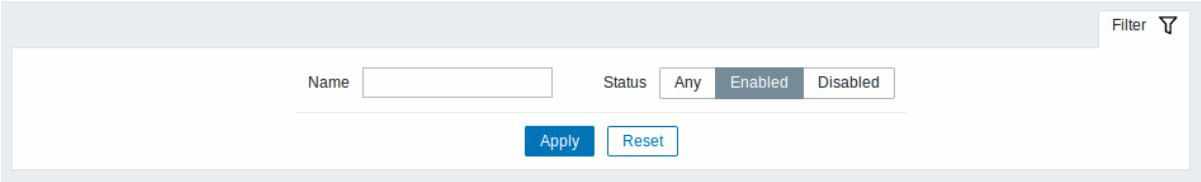
- *Включить* - изменить статус правила обнаружения на *Включено*
- *Отключить* - изменить статус правила обнаружения на *Отключено*
- *Удалить* - удалить правила обнаружения

Чтобы использовать эти параметры, отметьте флажки перед соответствующими правилами обнаружения, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила обнаружения, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком правил обнаружения. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать правила обнаружения по имени и статусу.



Оповещения

Обзор

В этом меню есть разделы, связанные с настройкой оповещений в Zabbix.

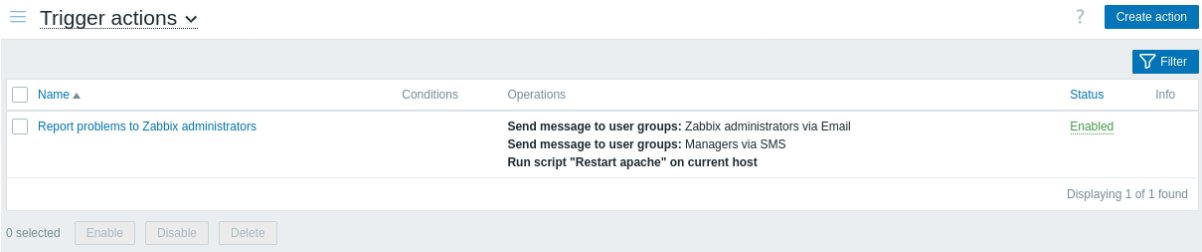
Действия

Обзор

В разделе *Оповещения* > *Действия* пользователи могут настраивать и обслуживать действия.

Отображаются действия, назначенные выбранному источнику событий (триггеры, сервисы, обнаружение, автоматическая регистрация, внутренние действия).

Чтобы переключиться на другой источник событий, нажмите *Действия* в разделе меню *Оповещения*. Также можно переключаться между источниками с помощью раскрывающегося списка заголовка в левом верхнем углу.



Отображаемые данные:

Столбец	Описание
Имя	Имя действия. При нажатии на имя действия открывается форма настройки действия.
Условия	Отображаются условия действия.
Операции	Отображаются операции действия. В списке операций также отображаются тип носителя (электронная почта, SMS или скрипт), используемый для отправки уведомления, а также имя и фамилия получателя уведомления (в скобках после имени пользователя). Операция действия может быть уведомлением или удалённой командой в зависимости от выбранного типа операции.
Состояние	Отображается состояние действия - *Включено* или *Отключено*. Нажав на состояние, его можно изменить. Дополнительные сведения о том, что происходит при отключении действия во время выполняемой эскалации, см. в разделе Эскалации.
Информация	Если всё работает правильно, в этом столбце значок не отображается. В случае ошибок, например отсутствия операций или условий действия после обновления или низкоуровневого обнаружения, отображается значок предупреждения. Наведите указатель мыши на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

Чтобы настроить новое действие, нажмите кнопку *Создать действие* в правом верхнем углу.

Для пользователей без прав супер-администратора действия отображаются в соответствии с настройками разрешений. Это означает, что в некоторых случаях пользователь без прав супер-администратора не может просмотреть полный список действий из-за определённых ограничений разрешений. Действие отображается пользователю без прав супер-администратора, если выполнены следующие условия:

- Пользователь имеет доступ на чтение и запись к группам узлов сети, узлам сети, шаблонам и триггерам в условиях действия
- Пользователь имеет доступ на чтение и запись к группам узлов сети, узлам сети и шаблонам в операциях действия, операциях восстановления и операциях обновления
- Пользователь имеет доступ на чтение к группам пользователей и пользователям в операциях действия, операциях восстановления и операциях обновления

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

- *Включить* - изменить статус действия на *Включено*
- *Отключить* - изменить статус действия на *Отключено*
- *Удалить* - удалить действия

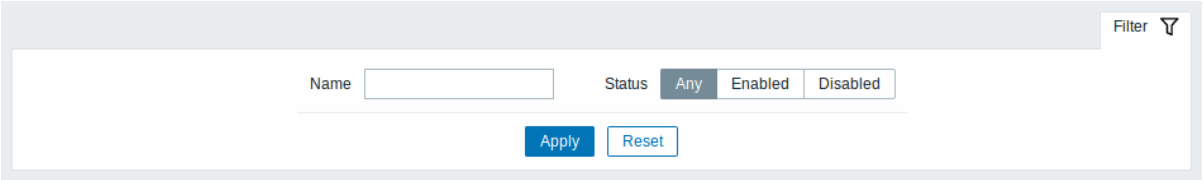
Чтобы использовать эти параметры, отметьте флажки перед соответствующими действиями, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те действия, которые вас интересуют.

Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Фильтр* доступна над списком действий. Если щелкнуть по ней, станет доступен фильтр, с помощью которого можно фильтровать действия по имени и статусу.



Типы медиа

Обзор

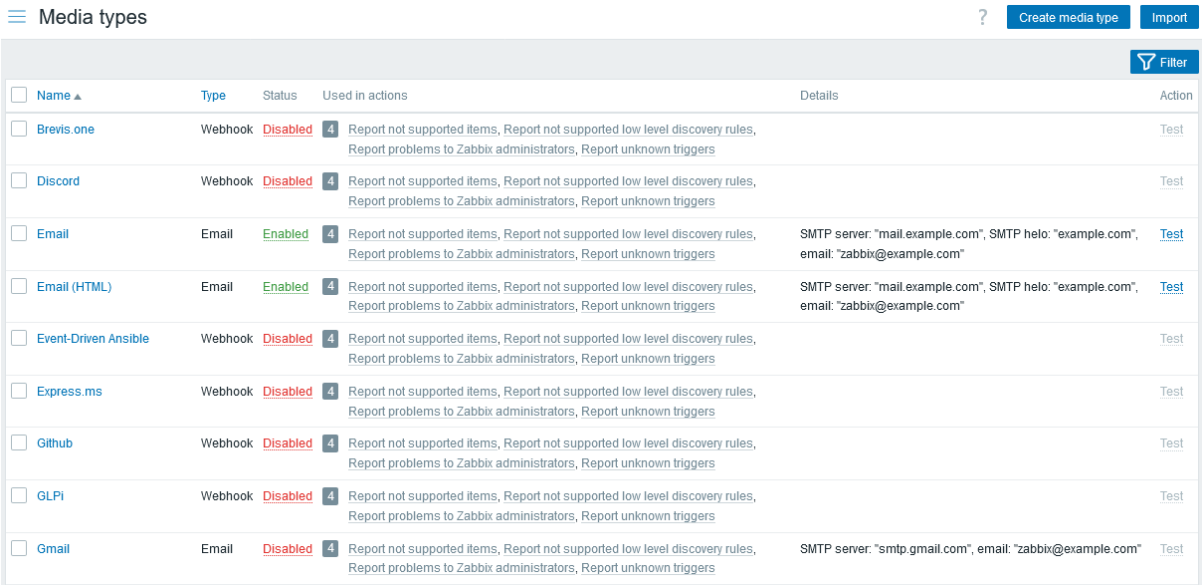
В разделе *Оповещения* > *Типы носителей* пользователи могут настраивать и обслуживать информацию о типах носителей.

Это меню доступно только пользователям типа SuperAdmin.

Информация о типе носителя содержит общие инструкции по использованию носителя в качестве канала доставки уведомлений. Конкретные сведения, например отдельные адреса электронной почты, на которые необходимо отправлять уведомление, хранятся у отдельных пользователей.

Отображается список существующих типов носителей с подробной информацией о них.

Note:
Если способы доставки (*Email*, *SMS*, *Webhook*, *Script* или *Push*) указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса, соответствующие предварительно настроенные типы носителей отключаются в этом разделе.



Отображаемые данные:

Столбец	Описание
Имя	Имя типа носителя. При нажатии на имя открывается форма настройки типа носителя.
Тип	Отображается тип носителя (электронная почта, SMS и т. д.).
Состояние	Отображается состояние типа носителя — *Включен* или *Отключен*. При нажатии на состояние его можно изменить.
Используется в действиях	Отображаются действия, в которых используется тип носителя, перед ними указано общее количество таких действий. При нажатии на имя действия открывается форма настройки действия. Если у пользователя нет разрешений на действие, имя нельзя нажать.
Подробности	Отображается подробная информация о типе носителя.
Действия	Доступно следующее действие: **Тест** — нажмите, чтобы открыть форму тестирования, в которой можно указать параметры типа носителя (например, адрес получателя, тему и текст тестового сообщения) и отправить тестовое сообщение, чтобы проверить работу настроенного типа носителя. См. также: Тестирование типа носителя.

Чтобы настроить новый тип носителя, нажмите кнопку *Создать тип носителя* в правом верхнем углу.

Чтобы импортировать тип носителя, нажмите кнопку *Импортировать* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

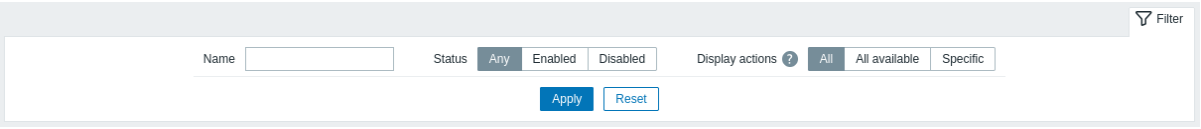
- *Включить* - изменить статус типа оповещений на *Включено*
- *Отключить* - изменить статус типа оповещений на *Отключено*
- *Экспорт* - экспортировать типы оповещений в файл YAML, XML или JSON
- *Удалить* - удалить типы оповещений

Чтобы использовать эти параметры, отметьте флажки перед соответствующими типами оповещений, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те типы медиа, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком типов медиа. Если щелкнуть ее, станет доступен фильтр, с помощью которого можно фильтровать типы медиа по имени и статусу. Кроме того, с помощью фильтра можно отображать действия в столбце *Used in actions* в зависимости от области использования их типа медиа (определяется параметром *Send to media type* в деталях операции действия).



Скрипты

Обзор

В разделе *Оповещения* > *Скрипты* можно настраивать и обслуживать пользовательские глобальные скрипты.

Это меню доступно только пользователям типа SuperAdmin.

Глобальные скрипты, в зависимости от настроенной области действия и разрешений пользователей, доступны для выполнения:

- Из меню узла сети в различных разделах веб-интерфейса (*Панель мониторинга*, *Проблемы*, *Последние данные*, *Карты* и т. д.).
- Из меню события.
- В качестве операции действия.

Скрипты выполняются агентом Zabbix, сервером Zabbix (прокси) или только сервером Zabbix. См. также раздел Выполнение команд.

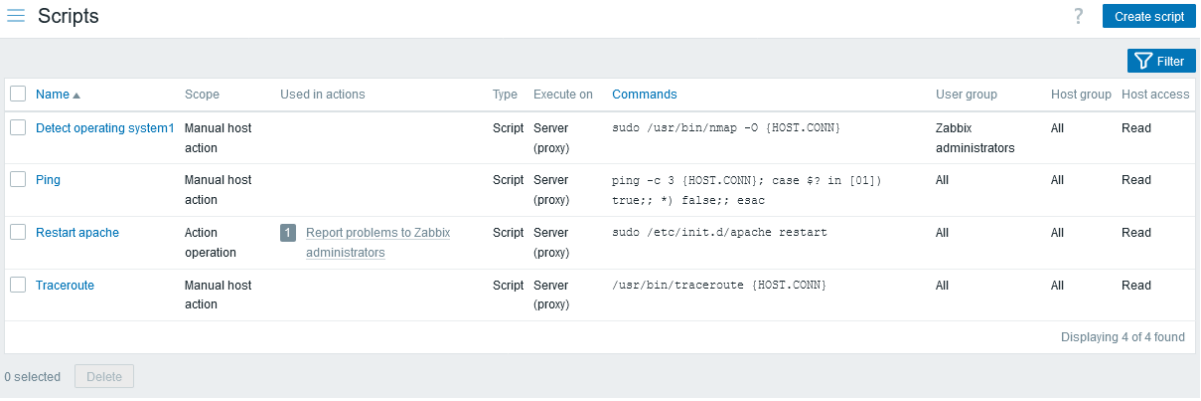
Удаленные команды по умолчанию отключены как на агенте Zabbix, так и на прокси Zabbix. Их можно включить следующим образом:

- Для удаленных команд, выполняемых агентом Zabbix, добавьте параметр `AllowKey=system.run[<command>,*]` или `AllowKeyRegexp=^system\.run\[<command>,.*\]$` для каждой разрешенной команды в конфигурации агента, где * соответствует любому режиму выполнения (`wait` или `nowait`).
- Для удаленных команд, выполняемых прокси Zabbix, установите значение параметра `EnableRemoteCommands` равным 1 в конфигурации прокси. Обратите внимание, что включать удаленные команды на прокси Zabbix не требуется, если они выполняются агентом Zabbix на узле сети, который отслеживается прокси.

Attention:
Не используйте `AllowKey=system.run[*]` или `AllowKeyRegexp=^system\.run\[.*\]$` без необходимости, поскольку это разрешает выполнение любой команды на узле сети. В целях безопасности рекомендуется разрешать только точные команды, а `system.run[*]` по умолчанию запрещен (`DenyKey=system.run[*]`).

Выполнение глобальных скриптов на сервере Zabbix можно отключить, установив `EnableGlobalScripts=0` в конфигурации сервера. В новых установках, начиная с Zabbix 7.0, выполнение глобальных скриптов на сервере Zabbix по умолчанию отключено.

Отображается список существующих скриптов с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя скрипта. При нажатии на имя скрипта открывается форма настройки скрипта.
Область действия	Область действия скрипта: операция действия, ручное действие для узла сети или ручное действие для события. Этот параметр определяет, где доступен скрипт: в удаленных командах операций действия, из меню узла сети или из меню события соответственно.
Используется в действиях	Отображаются все действия, в которых используется скрипт, перед ними указано общее количество таких действий. При нажатии на имя действия открывается форма настройки действия. Если у пользователя нет разрешений на действие, имя не является ссылкой.
Тип	Отображается тип скрипта: команда *URL*, *вебхук*, *скрипт*, *SSH*, *Telnet* или *IPMI*.
Выполнять на	Отображается, будет ли скрипт выполняться на агенте Zabbix, прокси Zabbix или сервере, либо только на сервере Zabbix.
Команды	Отображаются все команды, которые должны выполняться в рамках скрипта. Для вебхуков это поле остается пустым.
Группа пользователей	Отображается группа пользователей, для которой доступен скрипт (или *Все* для всех групп пользователей).
Группа узлов сети	Отображается группа узлов сети, для которой доступен скрипт (или *Все* для всех групп узлов сети).
Доступ к узлам сети	Отображается уровень разрешений для группы узлов сети: *Чтение* или *Запись*. Доступ к выполнению скрипта будет только у пользователей с необходимым уровнем разрешений.

Чтобы настроить новый скрипт, нажмите кнопку *Создать скрипт* в правом верхнем углу.

Параметры массового редактирования

Кнопка под списком предлагает один вариант массового редактирования:

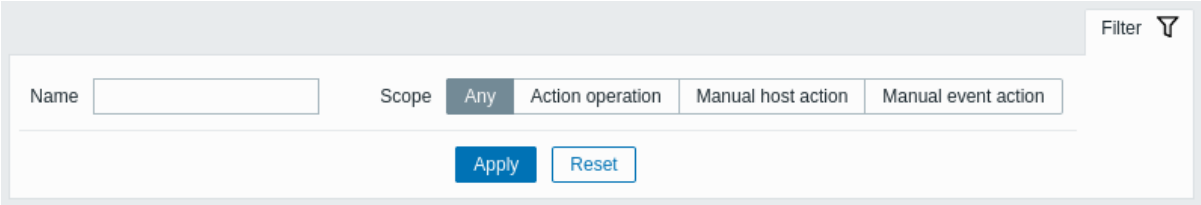
- *Удалить* - удалить скрипты

Чтобы использовать этот вариант, отметьте флажки перед соответствующими скриптами и нажмите *Удалить*.

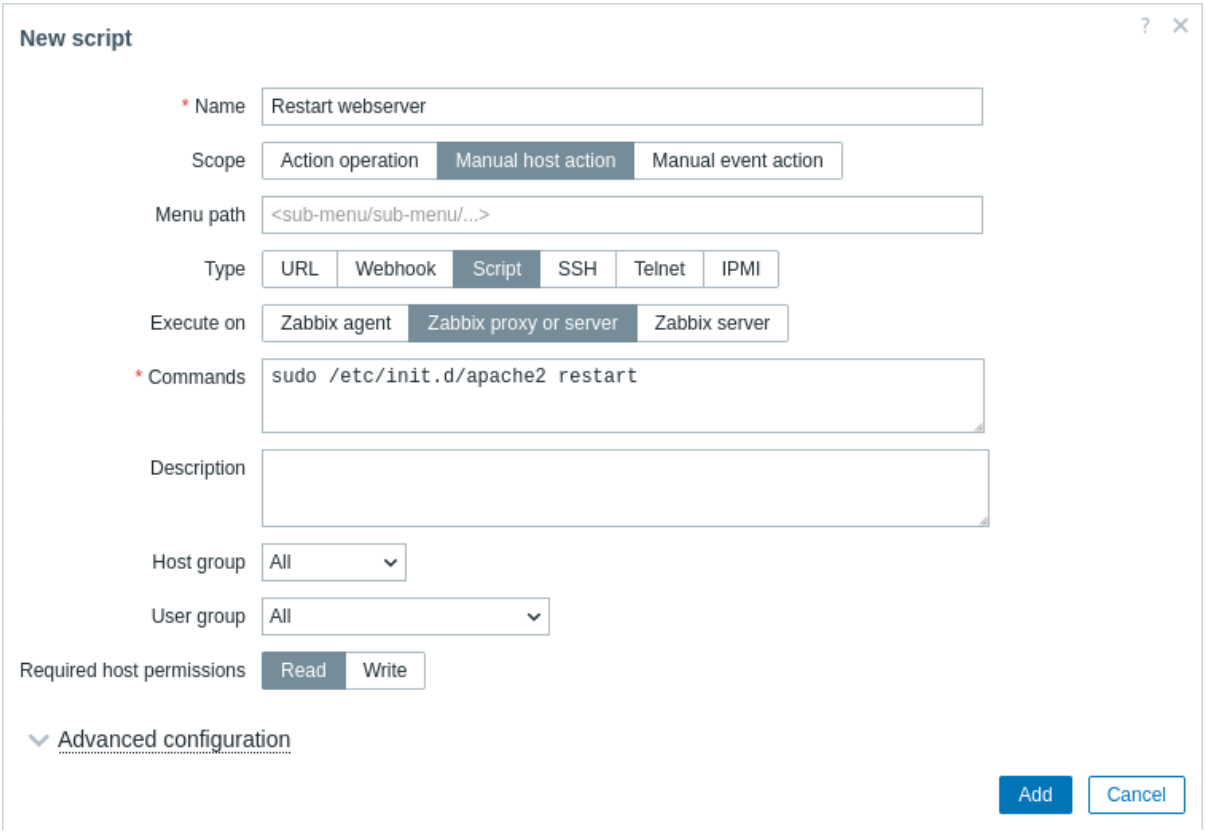
Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те скрипты, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком скриптов. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать скрипты по имени и области применения.



Настройка глобального скрипта



Note:
Для повышения безопасности в глобальных скриптах рекомендуется использовать функции макросов, а не обычные макросы, поскольку макросы не экранируются автоматически.

Атрибуты скрипта:

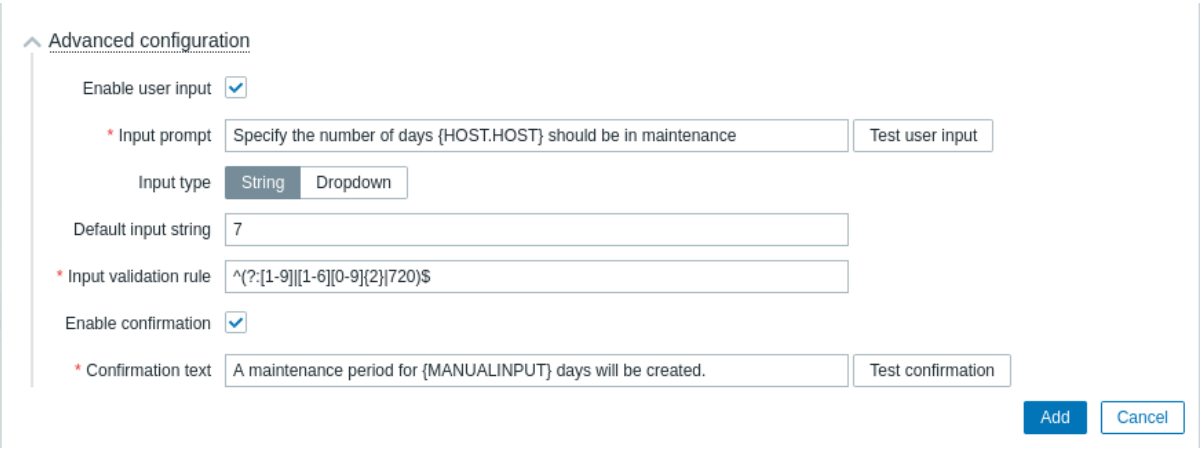
Параметр	Описание
Имя	Уникальное имя скрипта. Например, `/tmp`
Область действия	Область действия скрипта - операция действия, ручное действие над узлом сети или ручное действие над событием. Этот параметр определяет, где можно использовать скрипт: в удаленных командах операций действия, из меню узла сети или из меню события соответственно. Если для области действия установлено значение «Операция действия», скрипт доступен всем пользователям с доступом к разделу *Оповещения* > *Действия*. Если скрипт фактически используется в действии, его область действия нельзя изменить с «операции действия» на другую. **Поддержка макросов** Область действия влияет на набор доступных макросов. Например, связанные с пользователем макросы (`{USER.*}`) поддерживаются в скриптах, чтобы можно было передавать информацию о пользователе, запустившем скрипт. Однако они не поддерживаются, если область действия скрипта - операция действия, поскольку операции действий выполняются автоматически. Макрос `{MANUALINPUT}` позволяет указать ввод вручную во время выполнения скрипта. Он поддерживается в скриптах ручного действия над узлом сети и ручного действия над событием. Чтобы узнать, какие еще макросы поддерживаются, выполните поиск по запросам «Уведомления и команды на основе триггеров/Команды на основе триггеров», «Скрипты ручных действий над узлами сети» и «Скрипты ручных действий над событиями» в таблице поддерживаемых макросов. Обратите внимание: если макрос может преобразоваться в значение, содержащее пробелы (например, имя узла сети), не забудьте при необходимости заключить его в кавычки.

Параметр	Описание
Путь меню	Укажите требуемый путь меню к скрипту. Например, `Default` или `Default/` отобразит скрипт в соответствующем каталоге. Меню можно вкладывать друг в друга, например: `Main menu/Sub menu1/Sub menu2`. При доступе к скриптам через меню узла сети или события в разделах мониторинга они будут организованы в соответствии с указанными каталогами. Это поле отображается только если в качестве *Области действия* выбрано «Ручное действие над узлом сети» или «Ручное действие над событием».
Тип	Нажмите соответствующую кнопку, чтобы выбрать тип скрипта: команда **URL**, **вебхук**, **скрипт**, **SSH**, **Telnet** или **IPMI**. Тип **URL** доступен только если в качестве *Области действия* выбрано «Ручное действие над узлом сети» или «Ручное действие над событием».
Тип скрипта: **URL**	
URL	Укажите URL для быстрого доступа из меню узла сети или меню события. Поддерживаются макросы и пользовательские макросы пользователя. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). Используйте макрос {MANUALINPUT} в этом поле, чтобы можно было указать ввод вручную во время выполнения скрипта, например: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Значения макросов не должны быть закодированы в формате URL.
Открывать в новом окне	Определяет, следует ли открывать URL в новой или текущей вкладке браузера.
Тип скрипта: **вебхук**	
Параметры	Укажите переменные вебхука в виде пар «атрибут-значение». См. также: настройка медиа-типа вебхук. В значениях параметров поддерживаются макросы и пользовательские макросы пользователя. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше).
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при нажатии в поле параметра или на значок карандаша рядом с ним. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). См. также: настройка медиа-типа вебхук, Дополнительные объекты Javascript.
Тайм-аут	Тайм-аут выполнения JavaScript (1-60 с, по умолчанию 30 с). Поддерживаются суффиксы времени (например, 30s, 1m).
Тип скрипта: **скрипт**	
Выполнять на	Нажмите соответствующую кнопку, чтобы выполнить скрипт оболочки на: **агенте Zabbix** - скрипт будет выполнен агентом Zabbix (если элемент данных system.run разрешен) на узле сети; **прокси или сервере Zabbix** - скрипт будет выполнен прокси или сервером Zabbix в зависимости от того, прокси или сервер осуществляет мониторинг узла сети. Он будет выполнен на прокси, если это разрешено параметром EnableRemoteCommands. Он будет выполнен на сервере, если глобальные скрипты включены параметром сервера EnableGlobalScripts. **сервере Zabbix** - скрипт будет выполнен только сервером Zabbix. Этот параметр будет недоступен, если глобальные скрипты отключены параметром сервера EnableGlobalScripts.
Команды	Введите полный путь к командам, которые должны выполняться в скрипте. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). Поддерживаются пользовательские макросы пользователя.
Тип скрипта: **SSH**	
Метод аутентификации	Выберите метод аутентификации - пароль или открытый ключ.
Имя пользователя	Введите имя пользователя.
Пароль	Введите пароль. Это поле доступно, если в качестве метода аутентификации выбран «Пароль».
Файл открытого ключа	Введите путь к файлу открытого ключа. Это поле доступно, если в качестве метода аутентификации выбран «Открытый ключ».
Файл закрытого ключа	Введите путь к файлу закрытого ключа. Это поле доступно, если в качестве метода аутентификации выбран «Открытый ключ».
Парольная фраза	Введите парольную фразу. Это поле доступно, если в качестве метода аутентификации выбран «Открытый ключ».

Параметр	Описание
Порт	Введите порт удаленной службы SSH на целевом узле сети, к которому будет подключаться Zabbix.
Команды	Введите команды. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). Поддерживаются пользовательские макросы пользователя.
Тип скрипта: **Telnet**	
Имя пользователя	Введите имя пользователя.
Пароль	Введите пароль.
Порт	Введите порт удаленной службы Telnet на целевом узле сети, к которому будет подключаться Zabbix.
Команды	Введите команды. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). Поддерживаются пользовательские макросы пользователя.
Тип скрипта: **IPMI**	
Команда	Введите команду IPMI. Поддержка макросов зависит от области действия скрипта (см. *Область действия* выше). Поддерживаются пользовательские макросы пользователя.
Описание	Введите описание скрипта.
Группа узлов сети	Выберите группу узлов сети, для которой будет доступен скрипт (или *Все* для всех групп узлов сети).
Группа пользователей	Выберите группу пользователей, которой будет доступен скрипт (или *Все* для всех групп пользователей). Это поле отображается только если в качестве *Области действия* выбрано «Ручное действие над узлом сети» или «Ручное действие над событием».
Необходимые права на узел сети	Выберите уровень прав для группы узлов сети - *Чтение* или *Запись*. Только пользователи с необходимым уровнем прав смогут выполнять скрипт. Это поле отображается только если в качестве *Области действия* выбрано «Ручное действие над узлом сети» или «Ручное действие над событием».
Расширенная настройка	Нажмите заголовок *Расширенная настройка*, чтобы отобразить параметры расширенной настройки. Это поле отображается только если в качестве *Области действия* выбрано «Ручное действие над узлом сети» или «Ручное действие над событием».

Расширенная настройка

Дополнительные параметры настройки доступны в сворачиваемом разделе *Расширенная настройка*:



Параметр	Описание
Включить ввод пользователем	Установите флажок, чтобы включить ручной ввод пользователем перед выполнением скрипта. Ручной ввод пользователя заменит значение макроса {MANUALINPUT} в скрипте. См. также: Ручной ввод пользователя.

Параметр	Описание
Подсказка для ввода	Введите пользовательский текст с запросом на ввод данных. Этот текст будет отображаться над полем ввода во всплывающем окне *Ручной ввод*. Чтобы просмотреть всплывающее окно *Ручной ввод*, нажмите *Проверить ввод пользователя*. Предварительный просмотр также позволяет проверить, соответствует ли введенная строка правилу проверки ввода (см. параметры ниже). Поддержка макросов и пользовательских макросов зависит от области действия скрипта (см. *Область действия* в общих параметрах настройки скрипта).
Тип ввода	Выберите тип ручного ввода: **Строка** - одна строка; **Раскрывающийся список** - значение выбирается из нескольких вариантов раскрывающегося списка.
Варианты раскрывающегося списка	Введите уникальные значения для раскрывающегося списка пользовательского ввода через запятую. Чтобы добавить пустой вариант в раскрывающийся список, добавьте дополнительную запятую в начале, середине или конце списка. Это поле отображается только если для параметра *Тип ввода* выбрано значение «Раскрывающийся список».
Строка ввода по умолчанию	Введите строку пользовательского ввода по умолчанию (или оставьте поле пустым). Это поле будет проверено на соответствие регулярному выражению, указанному в поле *Правило проверки ввода*. Введенное здесь значение будет отображаться по умолчанию во всплывающем окне *Ручной ввод*. Это поле отображается только если для параметра *Тип ввода* выбрано значение «Строка».
Правило проверки ввода	Введите регулярное выражение для проверки строки пользовательского ввода. Поддерживаются глобальные регулярные выражения. Это поле отображается только если для параметра *Тип ввода* выбрано значение «Строка».
Включить подтверждение	Установите флажок, чтобы отображать сообщение с подтверждением перед выполнением скрипта. Эта функция может быть особенно полезна для потенциально опасных операций (например, скрипта перезагрузки) или операций, выполнение которых может занять много времени.
Текст подтверждения	Введите пользовательский текст подтверждения для всплывающего окна, включаемого с помощью расположенного выше флажка (например, *Удаленная система будет перезагружена. Вы уверены?*). Чтобы просмотреть, как будет выглядеть текст, нажмите *Проверить подтверждение* рядом с полем. Поддерживаются макросы и пользовательские макросы. *Примечание:* при проверке сообщения с подтверждением макросы не будут раскрыты.

Если настроены и ручной ввод пользователя, и сообщение с подтверждением, они будут отображаться в последовательных всплывающих окнах.

Ручной ввод пользователем

Ручной ввод пользователем позволяет указать пользовательский параметр при каждом выполнении скрипта. Это избавляет от необходимости создавать несколько похожих пользовательских скриптов, отличающихся только одним параметром.

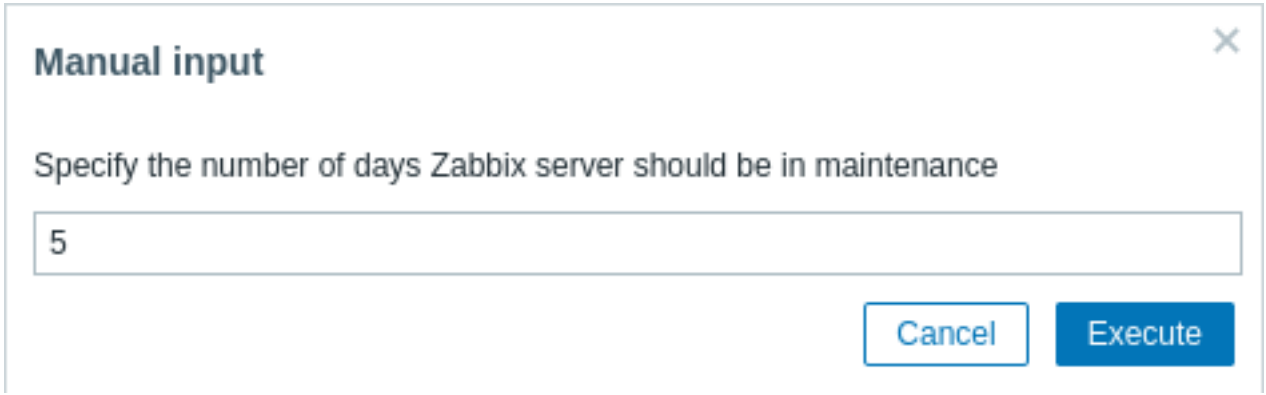
Например, при выполнении скрипта может потребоваться указать другое целое число или другой URL-адрес.

Чтобы включить ручной ввод пользователем:

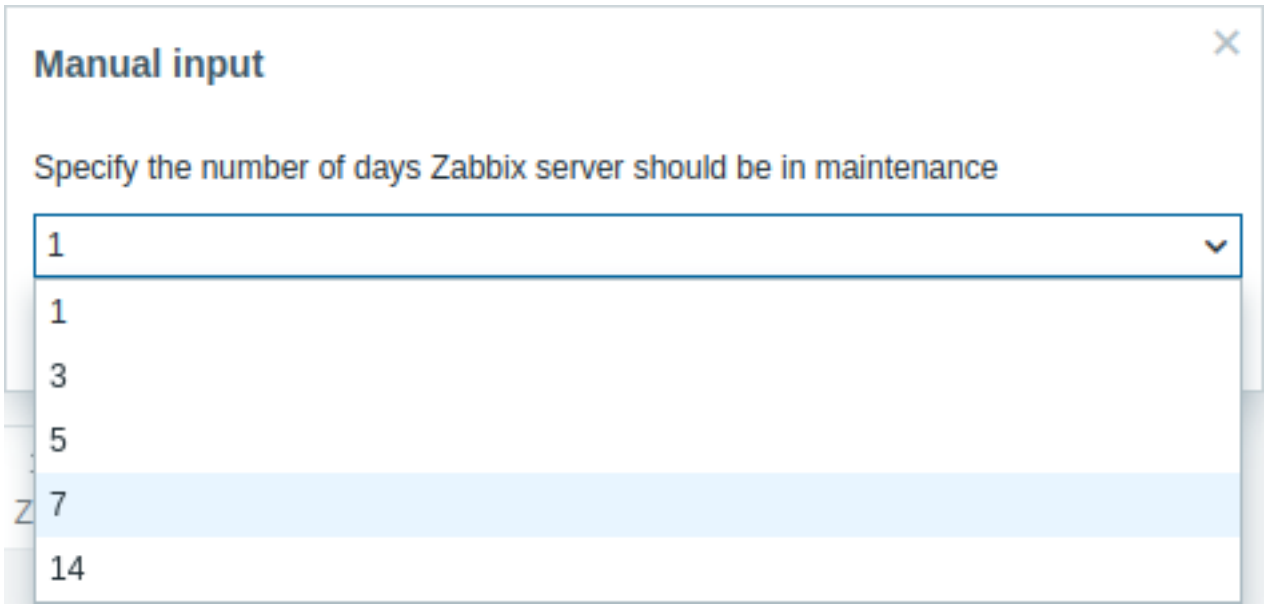
- используйте макрос {MANUALINPUT} в скрипте (командах, скрипте, параметре скрипта), где это необходимо; или в поле URL для скриптов URL;
- в разделе расширенной настройки скрипта включите ручной ввод пользователем и настройте параметры ввода.

Если ручной ввод включен, перед выполнением скрипта пользователю будет показано всплывающее окно *Ручной ввод* с предложением указать пользовательское значение. Указанное значение заменит {MANUALINPUT} в скрипте.

В зависимости от конфигурации пользователю будет предложено ввести строковое значение:



Или выбрать значение из раскрывающегося списка с заранее определенными вариантами:



Ручной ввод пользователем доступен только для скриптов, область действия которых имеет значение «Ручное действие над узлом сети» или «Ручное действие над событием».

Выполнение скрипта и результат

Скрипты, запускаемые сервером Zabbix, выполняются в порядке, описанном на странице выполнения команд.

Результат скрипта отображается во всплывающем окне, которое появляется после выполнения скрипта. Возвращаемое значение скрипта — это стандартный вывод:

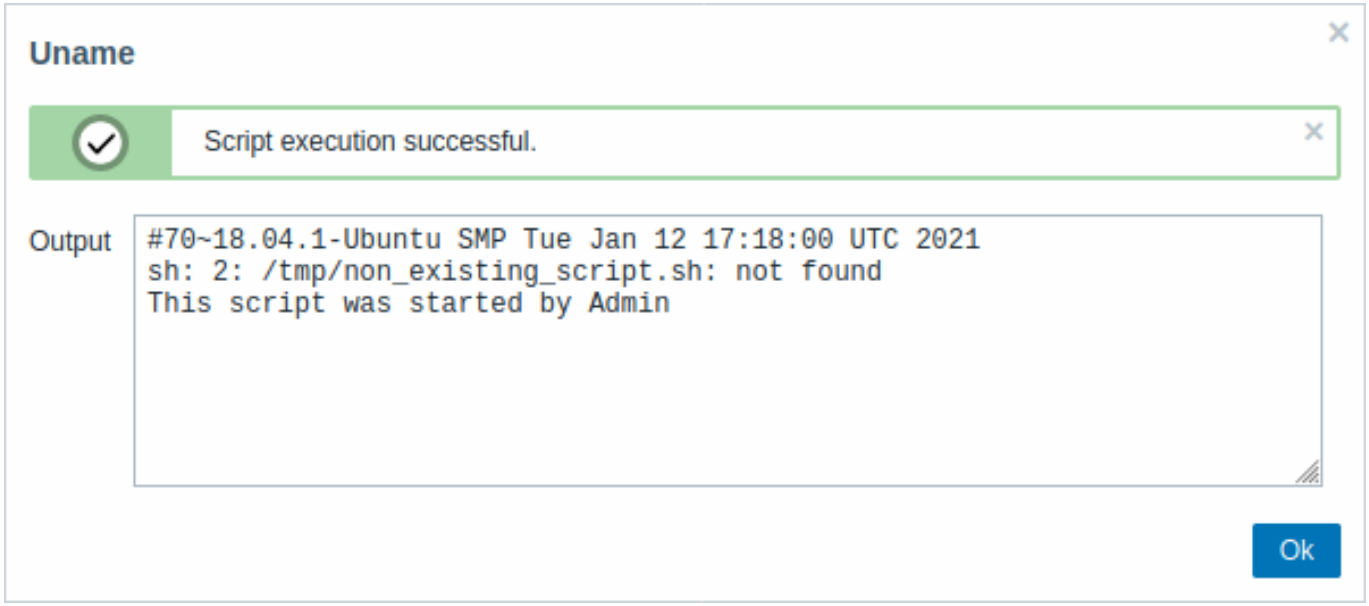
- Если скрипт завершается успешно (код выхода 0), возвращаемое значение ограничено 16 МБ (включая завершающие пробелы, которые обрезаются).
- Если скрипт завершается с ошибкой (ненулевой код выхода), возвращаемое значение стандартной ошибки ограничено 2 КБ.

По умолчанию Zabbix не сохраняет расширенный вывод скрипта. Чтобы сохранить полные сведения о выводе, вы можете реализовать ведение журнала непосредственно в самом скрипте (например, перенаправить вывод в локальный файл журнала).

Обратите внимание, что для скриптов, выполняемых либо на сервере Zabbix, либо на прокси Zabbix, также применяются ограничения базы данных.

Ниже приведен пример скрипта и окна результата:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



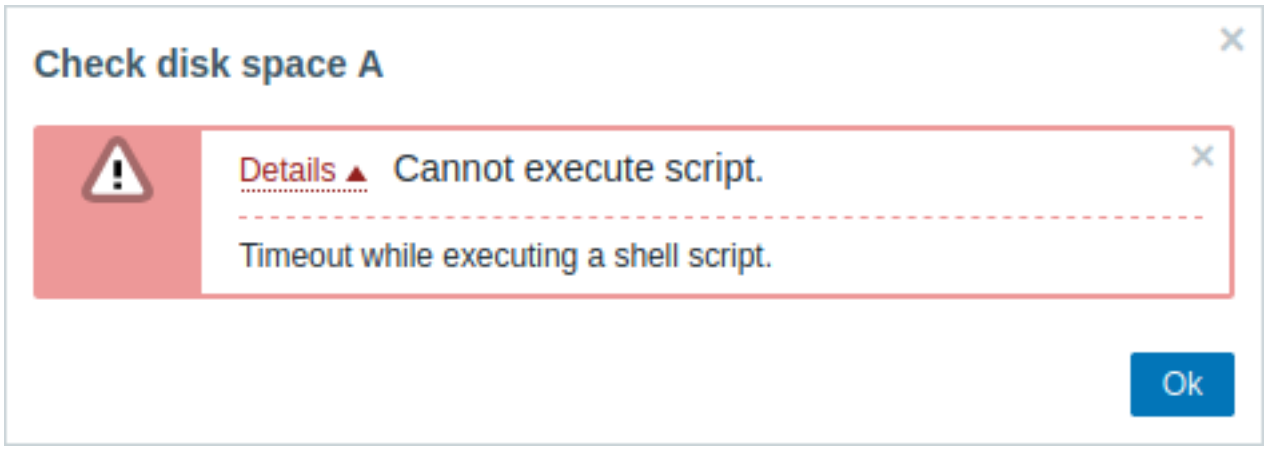
В результате выполнения скрипта сам скрипт не отображается.

Тайм-аут выполнения скрипта на агенте Zabbix

Вы можете столкнуться с ситуацией, когда во время выполнения скрипта возникает тайм-аут.

Ниже приведен пример скрипта, выполняемого на агенте Zabbix, и окна с результатом его выполнения:

```
sleep 5
df -h
```



В этом случае отображается следующее сообщение об ошибке:

```
Timeout while executing a shell script.
```

Чтобы избежать подобных ситуаций, рекомендуется оптимизировать сам скрипт (в приведенном выше примере — значение «5»), а не изменять параметр `Timeout` в конфигурации агента Zabbix и конфигурации сервера Zabbix.

Если параметр `Timeout` был изменен в конфигурации агента Zabbix, появится следующее сообщение об ошибке:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Это означает, что изменение было внесено в конфигурацию агента Zabbix, однако необходимо также изменить параметр `Timeout` в конфигурации сервера Zabbix.

Тайм-аут скрипта на активных агентах

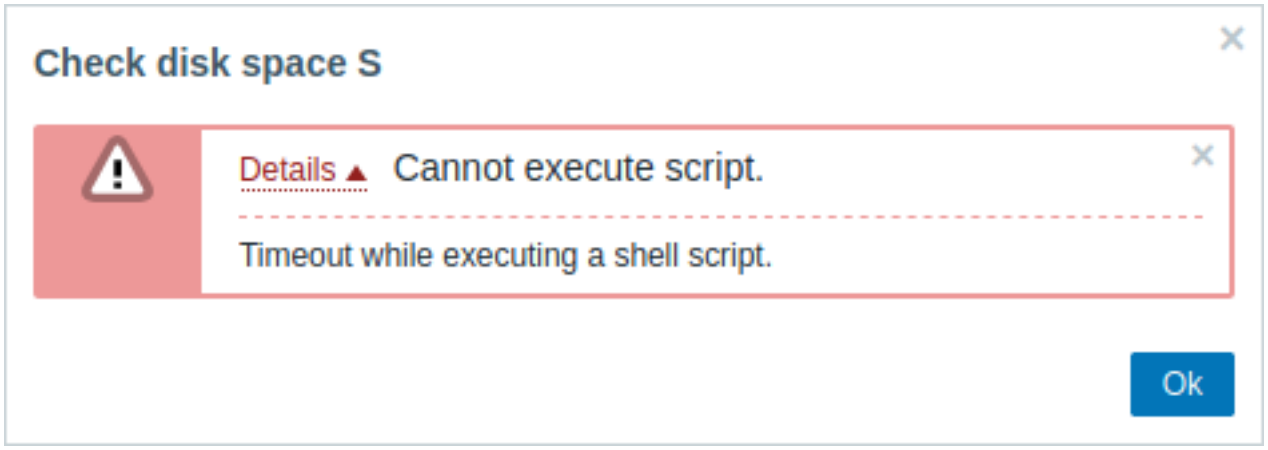
Возможно выполнение скриптов на агентах, работающих только в активном режиме. После того как выполнение скрипта будет инициировано операцией действия или ручным запуском скрипта, команда включается в конфигурацию активных проверок и выполняется, когда активный агент получает ее.

Ручные скрипты отправляются активному агенту вместе с тайм-аутом сервера/прокси для выполнения скрипта. Увеличьте значение тайм-аута по умолчанию для сервера/прокси для выполнения скрипта. Тайм-аут должен быть больше, чем частота обновления активных проверок, иначе тайм-аут будет превышен до того, как активный агент получит скрипт и сможет вернуть результат.

Тайм-аут скрипта на сервере/прокси Zabbix

Ниже приведен пример выполнения скрипта на сервере Zabbix и окно с результатом:

```
sleep 11
df -h
```



Также рекомендуется оптимизировать сам скрипт вместо настройки параметра `TrapperTimeout` на соответствующее значение (в нашем случае — > 11) путем изменения конфигурации сервера Zabbix.

Пользователи

Обзор

Это меню содержит разделы, связанные с настройкой пользователей в Zabbix. Это меню доступно только пользователям типа SuperAdmin.

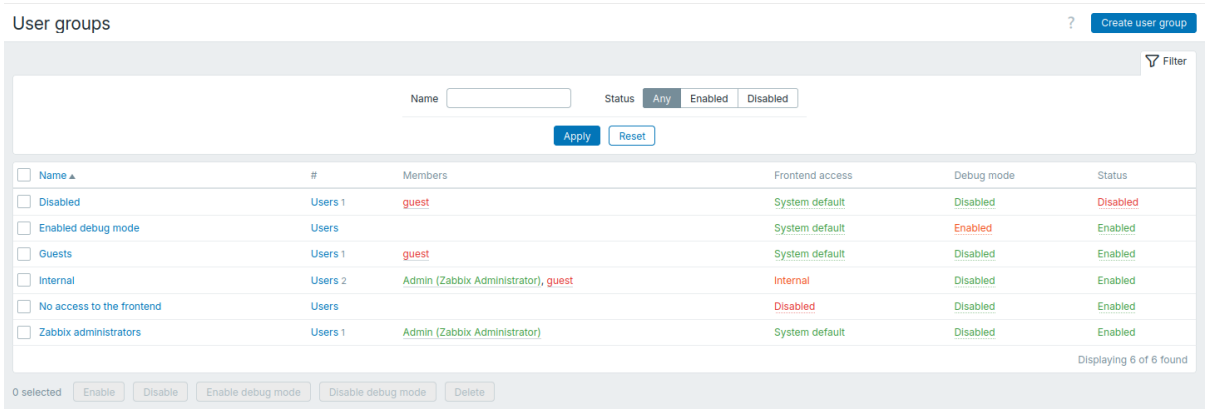
Группы пользователей

Обзор

В разделе *Users* > *User groups* ведется управление группами пользователей системы.

Группы пользователей

Отображается список существующих групп пользователей с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы пользователей. При нажатии на имя группы пользователей открывается форма настройки группы пользователей.
#	Количество пользователей в группе. При нажатии на *Пользователи* в списке пользователей будут отображены соответствующие пользователи с применением фильтра.

Столбец	Описание
Участники	Имена отдельных пользователей в группе пользователей (имя и фамилия указаны в скобках). При нажатии на имя пользователя открывается форма настройки пользователя. Пользователи из отключенных групп отображаются красным цветом.
Доступ к веб-интерфейсу	Отображается уровень доступа к веб-интерфейсу: **Системный по умолчанию** - пользователи проходят аутентификацию в Zabbix, LDAP или HTTP (в зависимости от метода аутентификации, заданного глобально); **Внутренняя** - пользователи проходят аутентификацию в Zabbix; игнорируется, если глобальным методом по умолчанию является аутентификация HTTP; **LDAP** - пользователи проходят аутентификацию через LDAP; игнорируется, если глобальным методом по умолчанию является аутентификация HTTP; **Отключен** - доступ к веб-интерфейсу Zabbix для этой группы запрещен. Нажав на текущий уровень, можно изменить его.
Режим отладки	Отображается состояние режима отладки - *Включен* или *Отключен*. Нажав на состояние, можно изменить его.
Состояние	Отображается состояние группы пользователей - *Включена* или *Отключена*. Нажав на состояние, можно изменить его.

Чтобы настроить новую группу пользователей, нажмите кнопку *Создать группу пользователей* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

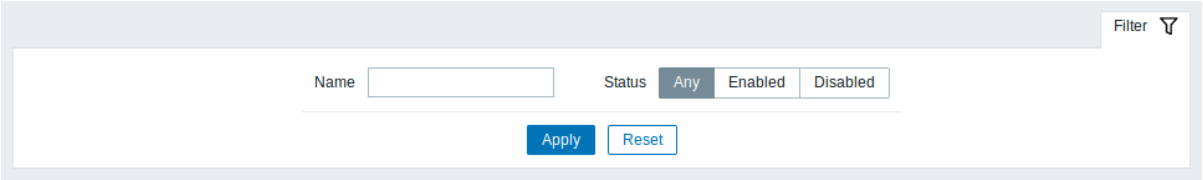
- *Включить* - изменить статус группы пользователей на *Включено*
- *Отключить* - изменить статус группы пользователей на *Отключено*
- *Включить режим отладки* - включить режим отладки для групп пользователей
- *Отключить режим отладки* - отключить режим отладки для групп пользователей
- *Удалить* - удалить группы пользователей

Чтобы использовать эти параметры, отметьте флажки перед соответствующими группами пользователей, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те группы пользователей, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

Ссылка *Filter* доступна над списком групп пользователей. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать группы пользователей по имени и статусу.



Роли пользователей

Обзор

В разделе *Пользователи* > *Роли пользователей* можно создавать роли пользователей.

Роли пользователей позволяют создавать детализированные разрешения на основе изначально выбранного типа пользователя (*Пользователь*, *Администратор*, *Суперадминистратор*).

После выбора типа пользователя ему предоставляются все доступные разрешения для этого типа (по умолчанию они отмечены).

Разрешения можно отозвать только из набора, доступного для выбранного типа пользователя; расширить их за пределы этого набора нельзя.

Флажки недоступных разрешений отображаются серым цветом; пользователи не смогут получить доступ к элементу даже при вводе прямого URL-адреса этого элемента в браузере.

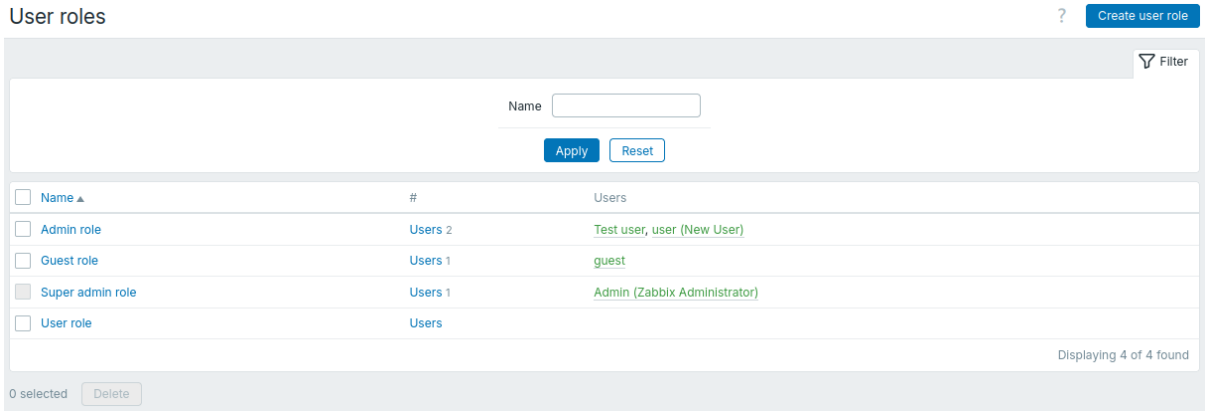
Note:
Ограничение доступа к некоторым элементам веб-интерфейса только предотвращает открытие соответствующей страницы — оно не лишает возможности получить доступ к базовым данным в других разделах интерфейса.

Роли пользователей можно назначать системным пользователям. Каждому пользователю можно назначить только одну роль.

Роли пользователей по умолчанию

По умолчанию Zabbix настроен с четырьмя ролями пользователей, для которых заранее определен набор разрешений:

- Роль гостя
- Роль пользователя
- Роль администратора
- Роль суперадминистратора



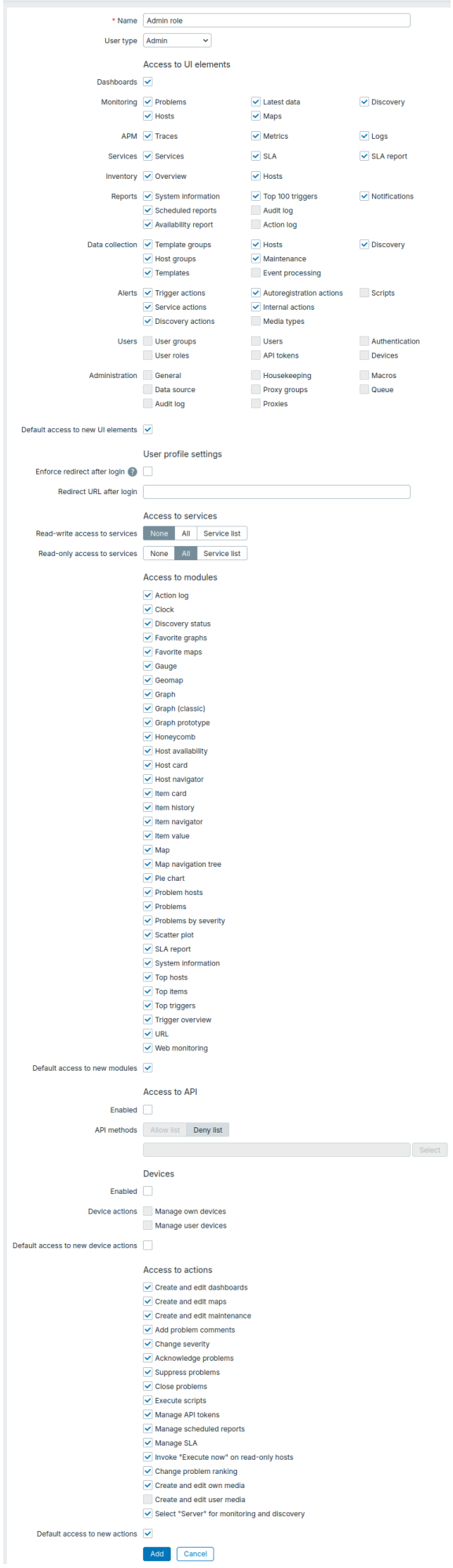
Они основаны на основных типах пользователей в Zabbix. Отображается список всех пользователей, которым назначена соответствующая роль. Пользователи, включенные в отключенные группы, отображаются красным цветом. *Роль гостя* — это роль типа «пользователь», которая предоставляет разрешения только на просмотр некоторых разделов веб-интерфейса.

Note:
Роль *суперадминистратора* по умолчанию нельзя изменить или удалить, поскольку в Zabbix должен существовать как минимум один пользователь типа «суперадминистратор» с неограниченными привилегиями. Пользователи типа *суперадминистратор* могут изменять настройки своей роли, но не ее тип.

Настройка

Чтобы создать новую роль, нажмите кнопку *Создать роль пользователя* в правом верхнем углу. Чтобы обновить существующую роль, нажмите ее название, чтобы открыть форму настройки.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

Отображаются доступные разрешения. Чтобы отозвать определенное разрешение, снимите соответствующий флажок.

Ниже описаны доступные разрешения и значения по умолчанию для каждой предварительно настроенной роли пользователя в Zabbix.

Разрешения по умолчанию

Доступ к элементам веб-интерфейса

Доступ к разделам меню по умолчанию зависит от типа пользователя. Подробнее см. на странице «Разрешения» здесь.

Доступ к другим параметрам

Некоторые разделы, описанные в таблице ниже, могут быть скрыты:

- *Доступ к модулям* скрыт, если в файле конфигурации веб-интерфейса для `$ZBX_FEATURE_FLAGS['modules_config_enabled` установлено значение `false`.
- *Устройства* скрыты, если параметр `EnableMobileDevices` отключен.

Параметр	**Описание**	**Роли пользователей по умолчанию**			
		Роль супер-администратора	*Роль администратора*	*Роль пользователя*	*Роль гостя*
Доступ по умолчанию к новым элементам веб-интерфейса	Этот параметр определяет, как будут доступны новые разделы меню после обновления Zabbix. Существующие разделы меню модулей не изменяются.	Да	Да	Да	Да
Настройки профиля пользователя					
Принудительно перенаправлять после входа	Установите флажок, чтобы ограничить пользователей URL-адресом перенаправления после входа. Если флажок не установлен, пользователи могут задать личный URL-адрес после входа в своем профиле; URL-адрес перенаправления после входа используется только как резервный.	Да	Нет	Нет	Нет
URL-адрес перенаправления после входа	Укажите URL-адрес перенаправления после входа.				
Доступ к сервисам					
Доступ к сервисам с правом чтения и записи	Выберите доступ к сервисам с правом чтения и записи: **Нет** — полный запрет доступа **Все** — доступ с правом чтения и записи ко всем сервисам **Список сервисов** — выбор сервисов, к которым предоставляется доступ с правом чтения и записи Предоставленный доступ с правом чтения и записи имеет приоритет над настройками доступа только для чтения и динамически наследуется дочерними сервисами.	Все	Все	Нет	Нет

Доступ к сервисам с правом чтения и записи по тегу	Укажите имя тега и, при необходимости, значение, чтобы дополнительно предоставить доступ с правом чтения и записи к сервисам, соответствующим тегу. Этот параметр доступен, если в параметре *Доступ к сервисам с правом чтения и записи* выбран вариант *Список сервисов*. Предоставленный доступ с правом чтения и записи имеет приоритет над настройками доступа только для чтения и динамически наследуется дочерними сервисами.				
Доступ к сервисам только для чтения	Выберите доступ к сервисам только для чтения: **Нет** — полный запрет доступа **Все** — доступ только для чтения ко всем сервисам **Список сервисов** — выбор сервисов, к которым предоставляется доступ только для чтения Доступ только для чтения не имеет приоритета над доступом с правом чтения и записи и динамически наследуется дочерними сервисами.			Все	Все
Доступ к сервисам только для чтения по тегу	Укажите имя тега и, при необходимости, значение, чтобы дополнительно предоставить доступ только для чтения к сервисам, соответствующим тегу. Этот параметр доступен, если в параметре *Доступ к сервисам только для чтения* выбран вариант *Список сервисов*. Доступ только для чтения не имеет приоритета над доступом с правом чтения и записи и динамически наследуется дочерними сервисами.				
Доступ к модулям					
<Имя модуля>	Разрешить или запретить доступ к определенному модулю. В этом разделе отображаются только включенные модули. Невозможно предоставить или ограничить доступ к модулю, который в данный момент отключен.	Да	Да	Да	Да
Доступ по умолчанию к новым модулям	Этот параметр определяет, как будут доступны новые модули и виджеты после обновления Zabbix. Он также применяется к модулям и виджетам, добавленным в разделе *Администрирование > Общее > Модули*.				
Доступ к API					
Включено	Включить или отключить доступ к API.	Да	Нет	Нет	Нет

Методы API	Выберите *Список разрешений*, чтобы разрешить указанные в поле поиска методы API, или *Список запретов*, чтобы запретить их. Обратите внимание, что невозможно одновременно разрешить одни методы API и запретить другие. Начните вводить имя метода в поле поиска, затем выберите метод из списка автодополнения. Также можно нажать кнопку *Выбрать* и выбрать методы из полного списка, доступного для этого типа пользователя. Обратите внимание, что если определенное действие в блоке «Доступ к действиям» не выбрано, пользователи не смогут использовать методы API, связанные с этим действием. Поддерживаются подстановочные знаки. Примеры: `dashboard.*` (все методы сервиса API «dashboard.»), `*` (любой метод), `*.export` (методы с именем «.export» из всех сервисов API). Если методы не указаны, *Список разрешений* запрещает доступ ко всем методам API, а *Список запретов* разрешает доступ ко всем методам API, доступным для данного типа пользователя. Чтобы разрешить все методы с помощью *Списка разрешений*, укажите `*`. Чтобы полностью заблокировать доступ к API, снимите флажок *Включено*, а не оставляйте *Список разрешений* пустым. Список методов API не применяется к методам, не требующим аутентификации, например `apiinfo.version` и `user.login`.				
Доступ к действиям					
Создание и редактирование панелей Создание и редактирование карт	Снятие этого флажка также отменяет права на использование методов API `.create`, `.update` и `.delete` для соответствующих элементов.	Да	Да	Да	Нет
Создание и редактирование обслуживания				Нет	
Добавление комментариев к проблемам Изменение серьезности Подтверждение проблем Подавление проблем Закрытие проблем	Снятие этого флажка также отменяет права на выполнение соответствующего действия с помощью метода API `event.acknowledge`.			Да	
Выполнение скриптов	Снятие этого флажка также отменяет права на использование метода API `script.execute`.				
Управление токенами API	Снятие этого флажка также отменяет права на использование всех методов API `token.`.				

Управление запланированными отчетами	Снятие этого флажка также отменяет права на использование всех методов API `report.*`.			Нет	
Управление SLA	Включить или отключить права на управление SLA.				
Вызов «Выполнить сейчас» на узлах сети только для чтения	Разрешить пользователям использовать параметр «Выполнить сейчас» в разделе «Последние данные» для элементов данных на узлах сети только для чтения.			Да	
Изменение ранжирования проблем	Разрешить пользователям изменять ранжирование проблемы с причины на симптом и наоборот.				
Создание и редактирование собственных каналов	Разрешить пользователям создавать и редактировать собственные каналы.				
Создание и редактирование каналов пользователей	Разрешить пользователям создавать и редактировать каналы других пользователей. Этот параметр доступен только пользователям с ролью супер-администратора.		Нет	Нет	
Выбор «Сервер» для мониторинга и обнаружения	Снятие этого флажка ограничивает создание узлов сети, мониторируемых сервером, и правил обнаружения сети, выполняемых сервером, а также тестирование элементов данных с помощью сервера. Параметр *Сервер* будет отключен в настройке узла сети для параметра *Мониторируется*, в настройке правила обнаружения сети для параметра *Обнаружение с помощью* и в тестировании элементов данных для параметра *Тестировать с помощью*.		Да	Нет	
Доступ по умолчанию к новым действиям	Этот параметр определяет, как будут доступны новые действия после обновления Zabbix.		Да	Да	
Устройства					
Включено	Включить или отключить доступ к разделу *Настройки пользователя* > *Устройства* (доступен всем пользователям) и *Пользователи* > *Устройства* (доступен только супер-администраторам).	Да	Нет	Нет	Нет
Управление собственными устройствами	Разрешить пользователям управлять собственными устройствами в разделе *Настройки пользователя* > *Устройства*.				
Управление устройствами пользователей	Разрешить пользователям управлять устройствами, принадлежащими другим пользователям, в разделе *Пользователи* > *Устройства*.				
Доступ по умолчанию к новым действиям	Этот параметр определяет, как будут доступны новые действия после обновления Zabbix.				

См. также:

- Настройка пользователя

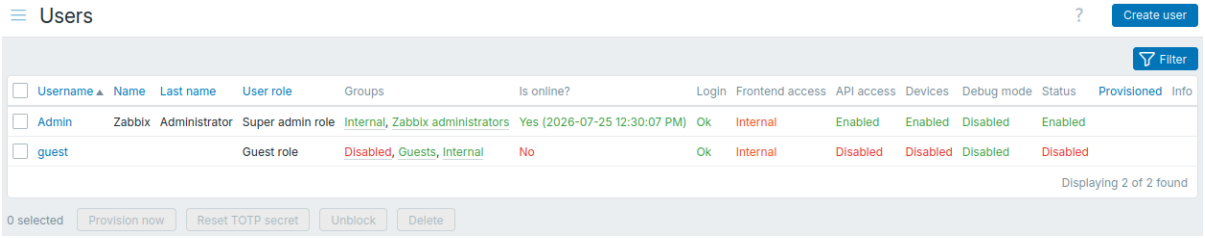
Пользователи

Обзор

В разделе *Пользователи* > *Пользователи* выполняется управление пользователями системы.

Пользователи

Отображается список существующих пользователей с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя пользователя	Имя пользователя для входа в Zabbix. При нажатии на имя пользователя открывается форма настройки пользователя.
Имя	Имя пользователя.
Фамилия	Фамилия пользователя.
Роль пользователя	Отображается роль пользователя.
Группы	Перечислены группы, в которые входит пользователь. При нажатии на имя группы пользователей открывается форма настройки группы пользователей. Отключенные группы отображаются красным цветом.
В сети?	Отображается статус пользователя - *Да* или *Нет*. В скобках отображается время последней активности пользователя.
Вход	Отображается статус входа пользователя - *OK* или *Заблокирован*. Пользователь может быть временно заблокирован после превышения количества неудачных попыток входа, заданного в разделе *Администрирование > Общие > Прочее* (по умолчанию - пять). Нажмите *Заблокирован*, чтобы разблокировать пользователя.
Доступ к веб-интерфейсу	Отображается уровень доступа к веб-интерфейсу - *Системный по умолчанию*, *Внутренний*, *LDAP* или *Отключен*, в зависимости от настройки для всей группы пользователей.
Доступ к API	Отображается статус доступа к API - *Включен* или *Отключен*, в зависимости от настройки для роли пользователя. Если доступ к API включен, при наведении указателя мыши на *Включен* отображаются методы API роли пользователя в разделе *Разрешенные методы* или *Запрещенные методы* в зависимости от режима методов API. Если методы не указаны, отображается *Нет*.
Устройства	Отображается статус доступа к устройствам - *Включен* или *Отключен*, в зависимости от настройки для роли пользователя. Этот столбец отображается, если включен параметр `EnableMobileDevices`.
Режим отладки	Отображается статус режима отладки - *Включен* или *Отключен*, в зависимости от настройки для всей группы пользователей.
Статус	Отображается статус пользователя - *Включен* или *Отключен*, в зависимости от настройки для всей группы пользователей.
Подготовлен	Отображается дата последней подготовки пользователя. Используется для пользователей, созданных с помощью JIT-подготовки из LDAP/SAML.
Информация	Отображается информация об ошибках. Для пользователей без групп пользователей отображается желтое предупреждение. Для пользователей без ролей, а также для пользователей без ролей и групп пользователей отображается красное предупреждение.

Чтобы настроить нового пользователя, нажмите кнопку *Создать пользователя* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком позволяют выполнять следующие операции массового редактирования:

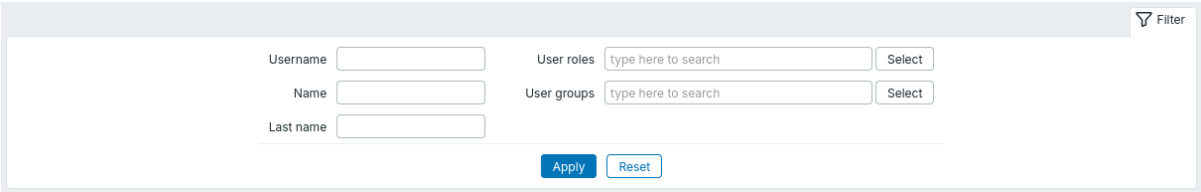
- *Предоставить сейчас* - обновить информацию о пользователях из LDAP (этот параметр доступен только в том случае, если выбран пользователь LDAP)
- *Сбросить секрет TOTP* - сбросить секреты TOTP пользователей для всех методов TOTP и удалить сеанс пользователя (этот параметр доступен только в том случае, если включена MFA; сеанс пользователей без секретов TOTP удален не будет)
- *Разблокировать* - повторно разрешить доступ к системе заблокированным пользователям
- *Удалить* - удалить пользователей

Чтобы воспользоваться этими параметрами, установите флажки перед соответствующими пользователями, а затем нажмите нужную кнопку.

Использование фильтра

С помощью фильтра можно отображать только интересующих вас пользователей. Для повышения производительности поиска данные ищутся с нераскрытыми макросами.

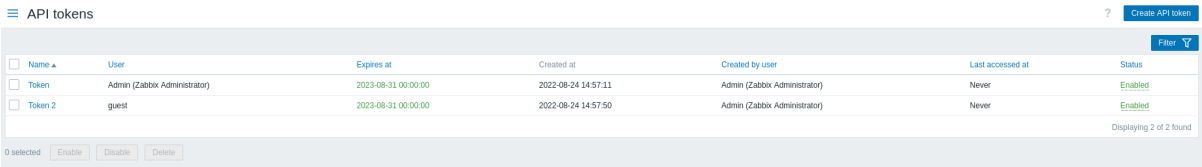
Ссылка *Фильтр* доступна над списком пользователей. При нажатии на нее становится доступен фильтр, с помощью которого можно фильтровать пользователей по имени пользователя, имени, фамилии, роли пользователя и группе пользователей.



API-токены

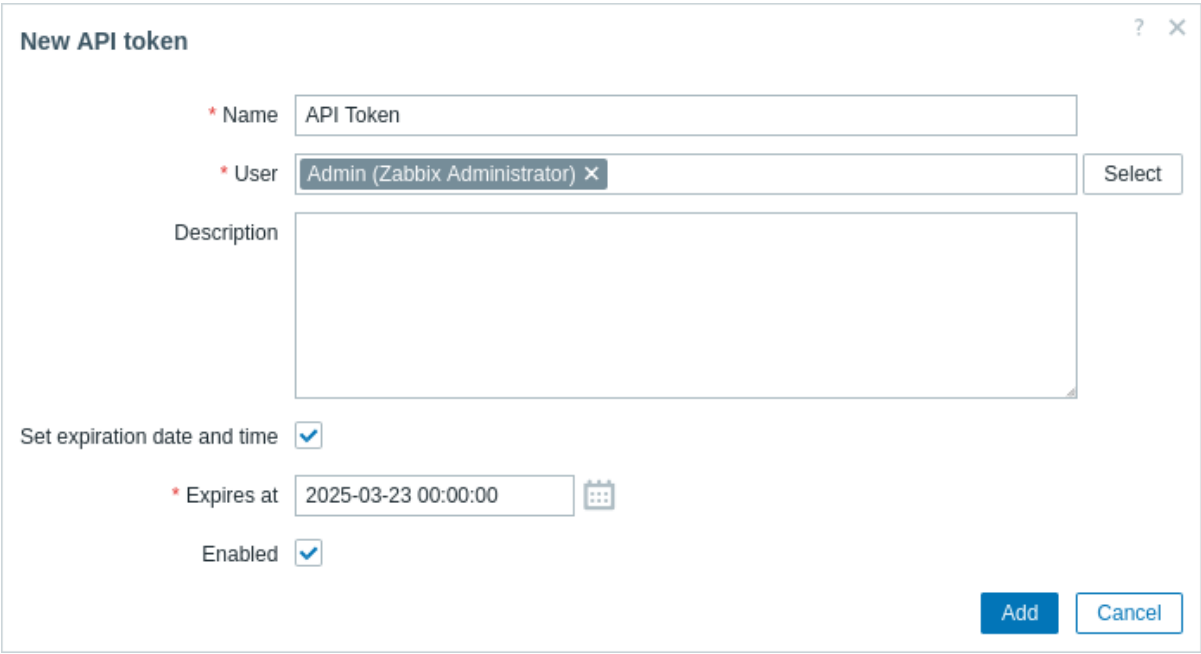
Обзор

В этом разделе можно создавать токены API и управлять ими.
 Токены API используются для аутентификации в API Zabbix.
 Каждый токен API назначается одному пользователю и наследует его права доступа. Запросы, аутентифицированные с помощью токена API, авторизуются в соответствии с ролью и правами связанного пользователя.



Вы можете фильтровать токены API по имени, пользователям, которым назначены токены, сроку действия, пользователям, создавшим токены, или статусу (включен/отключен). Нажмите на статус токена в списке, чтобы быстро включить или отключить токен. Также можно массово включить или отключить токены, выбрав их в списке и нажав кнопки *Enable/Disable* под списком.

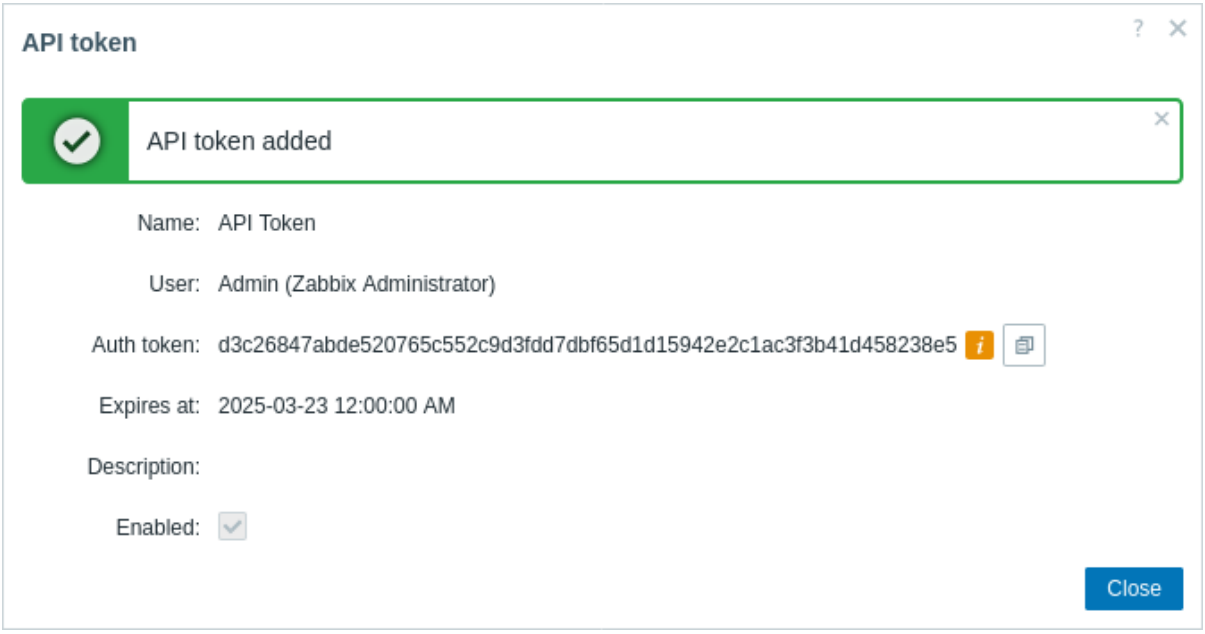
Чтобы создать новый токен, нажмите кнопку *Create API token* в правом верхнем углу, затем заполните обязательные поля на экране настройки токена:



Параметр	Описание
Name	Отображаемое имя токена.

Параметр	Описание
User	Пользователь, которому следует назначить токен. Чтобы быстро выбрать пользователя, начните вводить его имя пользователя, имя или фамилию, затем выберите нужного пользователя из списка автозаполнения. Также можно нажать кнопку *Select* и выбрать пользователя из полного списка пользователей. Токен можно назначить только одному пользователю.
Description	Необязательное описание токена.
Set expiration date and time	Снимите этот флажок, если у токена не должно быть срока действия.
Expiry date	Нажмите значок календаря, чтобы выбрать дату окончания действия токена, или введите дату вручную в формате YYYY-MM-DD hh:mm:ss
Enabled	Снимите этот флажок, если необходимо создать токен в отключенном состоянии.

Нажмите *Add*, чтобы создать токен.



Скопируйте значение *Auth token* и сохраните его в безопасном месте **до закрытия страницы**, затем нажмите *Close*. Токен появится в списке.

Warning:
Значение *Auth token* нельзя просмотреть повторно. Оно доступно только сразу после создания токена. Если вы потеряете сохраненный токен, его придется сгенерировать заново, при этом будет создана новая строка авторизации.

Нажмите имя токена, чтобы изменить его имя, описание, настройки срока действия или статус. Обратите внимание, что изменить пользователя, которому назначен токен, невозможно. Нажмите кнопку *Update*, чтобы сохранить изменения. Если токен был утерян или раскрыт, можно нажать кнопку *Regenerate*, чтобы сгенерировать новое значение токена. Появится диалоговое окно подтверждения с предложением подтвердить операцию, поскольку после ее выполнения ранее сгенерированный токен станет недействительным.

Пользователи без доступа к разделу меню *Administration* могут просматривать и изменять сведения о назначенных им токенах в разделе *User profile* > *API tokens* section только в том случае, если в разрешениях их роли пользователя разрешено *Manage API tokens*.

Аутентификация

Обзор

Раздел *Пользователи* > *Аутентификация* позволяет указать метод аутентификации пользователей в Zabbix и внутренние требования к паролям.

Доступны следующие методы аутентификации: внутренняя, HTTP, LDAP, SAML и MFA-аутентификация.

Аутентификация по умолчанию

По умолчанию Zabbix использует **внутреннюю** аутентификацию Zabbix для всех пользователей.

Можно изменить метод аутентификации по умолчанию на систему LDAP для всего экземпляра Zabbix. Для этого перейдите на вкладку *LDAP* и настройте параметры LDAP, затем вернитесь на вкладку *Аутентификация* и переключите селектор *Аутентификация по умолчанию* на LDAP.

Обратите внимание, что метод аутентификации можно дополнительно настроить на уровне группы пользователей. Даже если аутентификация LDAP настроена глобально, некоторые группы пользователей могут по-прежнему проходить аутентификацию в Zabbix. Для этих групп необходимо установить доступ к веб-интерфейсу — «Внутренний».

Также можно включить аутентификацию LDAP только для определенных групп пользователей, если глобально используется внутренняя аутентификация. В этом случае сведения для аутентификации LDAP можно указать и использовать для определенных групп пользователей, для которых затем необходимо установить доступ к веб-интерфейсу — «LDAP». Если пользователь входит хотя бы в одну группу пользователей с аутентификацией LDAP, он не сможет использовать внутренний метод аутентификации.

Методы аутентификации HTTP, SAML 2.0 и MFA можно использовать в дополнение к методу аутентификации по умолчанию.

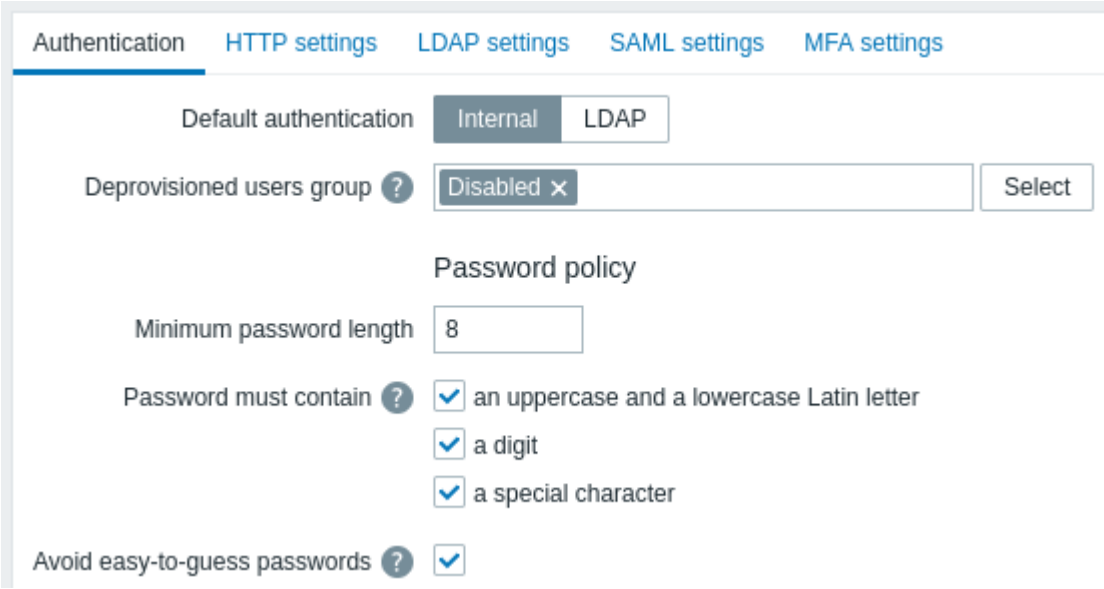
Zabbix поддерживает автоматическое создание учетных записей (JIT), которое позволяет создать учетные записи пользователей в Zabbix при первой аутентификации внешнего пользователя и настроить эти учетные записи. Автоматическое создание учетных записей поддерживается для LDAP и SAML.

См. также:

- Аутентификация HTTP
- Аутентификация LDAP
- Аутентификация SAML
- Аутентификация MFA

Конфигурация

На вкладке *Аутентификация* можно задать метод аутентификации по умолчанию, указать группу для депровижининга пользователей и установить требования к сложности паролей пользователей Zabbix.



Параметры конфигурации:

Параметр	Описание
Аутентификация по умолчанию	Выберите метод аутентификации по умолчанию для Zabbix: *Внутренняя* или *LDAP*.
Группа депровижининга пользователей	Укажите группу пользователей для пользователей, выведенных из депровижининга. Этот параметр требуется только для JIT-провижининга и относится к пользователям, созданным в Zabbix из систем LDAP или SAML, но больше не нуждающимся в провижининге. Необходимо указать отключенную группу пользователей.
Минимальная длина пароля	По умолчанию минимальная длина пароля составляет 8 символов. Поддерживаемый диапазон: 1-70. Обратите внимание, что пароли длиной более 72 символов будут усечены.

Параметр	Описание
Пароль должен содержать	Установите один или несколько флажков, чтобы потребовать использования в пароле указанных символов: - прописную и строчную латинские буквы - цифру - специальный символ Наведите указатель мыши на вопросительный знак, чтобы увидеть подсказку со списком символов для каждого параметра.
Запретить легко угадываемые пароли	Если этот параметр выбран, пароль будет проверен на соответствие следующим требованиям: - не должен содержать имя, фамилию или имя пользователя пользователя - не должен входить в список распространенных или контекстно-зависимых паролей. Список распространенных и контекстно-зависимых паролей автоматически формируется на основе списка NCSC «Top 100k passwords», списка SecLists «Top 1M passwords» и списка контекстно-зависимых паролей Zabbix. Внутренним пользователям не будет разрешено устанавливать пароли, включенные в этот список, поскольку такие пароли считаются слабыми из-за их широкого распространения.

Изменения требований к сложности паролей не повлияют на существующие пароли пользователей, однако если существующий пользователь решит изменить пароль, новый пароль должен будет соответствовать текущим требованиям. Подсказка со списком требований будет отображаться рядом с полем *Пароль* в профиле пользователя и в форме настройки пользователя, доступной из меню *Пользователи* > *Пользователи*.

HTTP

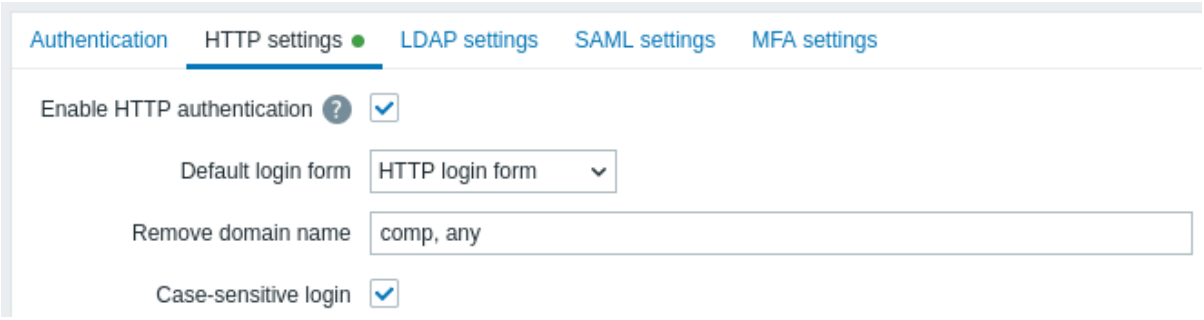
Обзор

Для проверки имен пользователей и паролей можно использовать аутентификацию на основе HTTP или веб-сервера (например, BasicAuthentication, NTLM/Kerberos). Обратите внимание, что пользователь также должен существовать в Zabbix, однако его пароль Zabbix использоваться не будет.

Attention:
Будьте осторожны! Перед включением этой функции убедитесь, что аутентификация веб-сервера настроена и работает правильно.

Note:
Аутентификацию HTTP можно отключить, установив `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` в файле конфигурации веб-интерфейса (`zabbix.conf.php`). В этом случае вкладка *Настройки HTTP* с параметрами аутентификации HTTP будет скрыта.

Конфигурация



Параметры конфигурации:

Параметр	Описание
Включить HTTP-аутентификацию	Установите флажок, чтобы включить HTTP-аутентификацию. При наведении указателя мыши на появится подсказка с предупреждением о том, что при аутентификации веб-сервера все пользователи (даже с доступом к веб-интерфейсу, настроенным как LDAP/Внутренняя) будут проходить аутентификацию на веб-сервере, а не в Zabbix.
Форма входа по умолчанию	Укажите, куда перенаправлять неаутентифицированных пользователей: **Форма входа Zabbix** — стандартная страница входа Zabbix. **Форма входа HTTP** — страница входа HTTP. Рекомендуется включать аутентификацию на основе веб-сервера только для страницы `index_http.php`. Если для параметра *Форма входа по умолчанию* установлено значение «Страница входа HTTP», пользователь будет автоматически авторизован, если модуль аутентификации веб-сервера установит действительное имя пользователя в переменной `$_SERVER`. Поддерживаются следующие ключи `$_SERVER`: `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Удалять имя домена	Список имен доменов, разделенных запятыми, которые следует удалять из имени пользователя. Например, `comp,any`: если имя пользователя — «Admin@any» или «comp\Admin», пользователь войдет в систему как «Admin»; если имя пользователя — «notacompany\Admin», вход будет запрещен.
Учитывать регистр при входе	Снимите флажок, чтобы отключить учет регистра в именах пользователей при входе (по умолчанию учет регистра включен). Отключение учета регистра позволяет, например, войти как «admin», даже если пользователь Zabbix называется «Admin» или «ADMIN». Обратите внимание: если учет регистра при входе отключен и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для этих пользователей всегда будет запрещен со следующим сообщением об ошибке: «Аутентификация не выполнена: предоставленные учетные данные не являются уникальными».

Note:
Для внутренних пользователей, которые не могут войти с использованием HTTP-учетных данных (если в качестве формы входа по умолчанию установлена форма входа HTTP), что приводит к ошибке 401, можно добавить строку `ErrorDocument 401 /index.php?form=default` к директивам базовой аутентификации. Это перенаправит пользователя на обычную форму входа Zabbix.

LDAP

Обзор

Внешний LDAP сервис authentication Может быть использован для проверки логинов и паролей пользователей

Аутентификация LDAP в Zabbix работает как минимум с Microsoft Active Directory и OpenLDAP.

Если настроен только вход через LDAP, пользователь также должен существовать в Zabbix, однако его пароль из Zabbix использоваться не будет. Если аутентификация проходит успешно, Zabbix сопоставляет локальное имя пользователя с атрибутом имени пользователя, возвращаемым LDAP.

Подготовка пользователей

Можно настроить JIT (just-in-time) **подготовку пользователей** для пользователей LDAP. В этом случае не требуется, чтобы пользователь уже существовал в Zabbix. Учетная запись пользователя может быть создана при первом входе пользователя в Zabbix.

Когда пользователь LDAP вводит свой логин и пароль LDAP, Zabbix проверяет, существует ли этот пользователь на *сервере* LDAP, заданном по умолчанию. Если пользователь существует, но у него еще нет учетной записи в Zabbix, в Zabbix создается новый пользователь, после чего пользователь может войти в систему.

Пользователь, созданный с помощью JIT-подготовки, связывается с сервером LDAP (каталогом), заданным по умолчанию на момент создания. Последующее изменение сервера LDAP по умолчанию не изменяет и не обновляет сервер LDAP, связанный с уже подготовленными пользователями.

Attention:
Если JIT-подготовка включена, на вкладке *Аутентификация* необходимо указать группу пользователей для депровижининга.

JIT-подготовка также позволяет обновлять подготовленные учетные записи пользователей на основе изменений в LDAP. Например, если пользователь перемещен из одной группы LDAP в другую, в Zabbix он также будет перемещен из одной группы в другую; если пользователь удален из группы LDAP, он также будет удален из группы в Zabbix и, если он не входит ни в одну другую группу, добавлен в группу пользователей для депровижининга. Подготовленные учетные записи пользователей обновляются в соответствии с настроенным периодом подготовки или при входе пользователя в Zabbix.

Обратите внимание, что фоновая подготовка выполняется веб-интерфейсом Zabbix, когда пользователь взаимодействует с ним или хотя бы имеет открытую страницу веб-интерфейса в браузере. Выделенных фоновых процессов для подготовки пользователей нет.

LDAP поддерживает три способа подключения к каталогу для аутентификации и поиска:

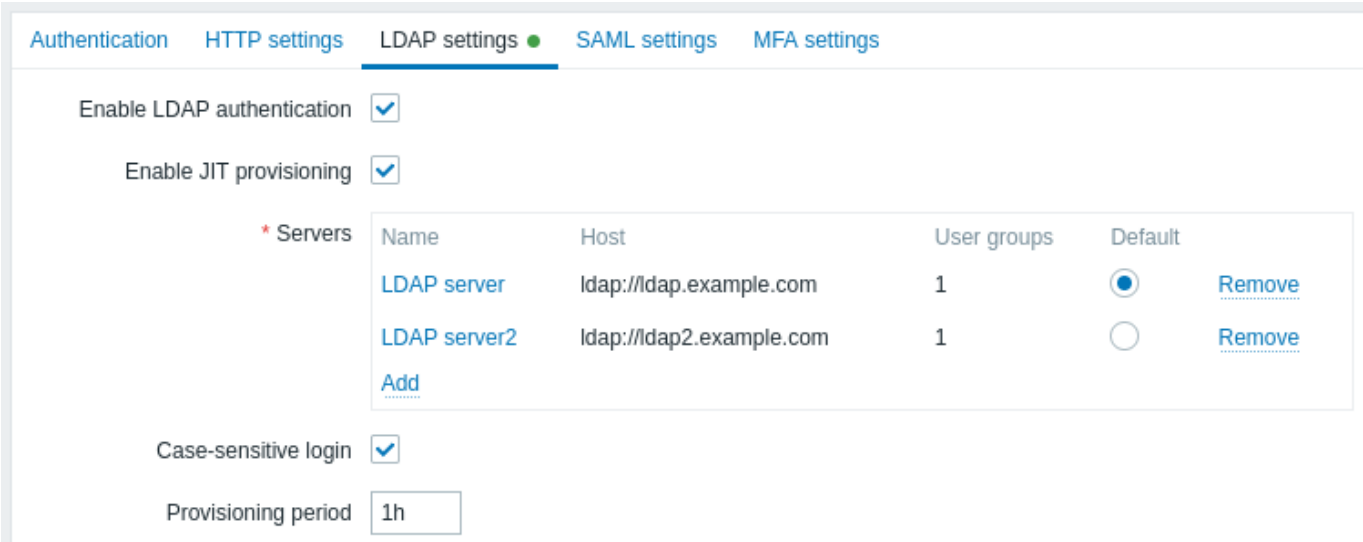
- Анонимное подключение - *Bind DN* / *Bind password* не указаны, а сервер LDAP разрешает анонимные запросы.
- Выделенный пользователь для подключения (сервисная учетная запись) - в *Bind DN* / *Bind password* указана конкретная учетная запись LDAP, которая используется Zabbix для поиска и подготовки пользователей. Это рекомендуемый и наиболее гибкий вариант, поскольку Zabbix может выполнять поиск и фоновую подготовку без учетных данных конечного пользователя.
- Прямое подключение пользователя - Zabbix выполняет подключение с использованием учетных данных, введенных пользователем при входе (параметры *Bind DN* / *Bind password* не настроены); этот режим настраивается путем добавления в Base DN заполнителя, например `uid=%{user}`. В этом режиме Zabbix получает доступ к паролю пользователя только во время интерактивного входа. В результате действия подготовки, требующие аутентификации в LDAP вне сеанса входа пользователя (например, с помощью кнопки *Подготовить сейчас* или при запуске фоновой подготовки, когда пользователь не вошел в систему), не могут пройти аутентификацию и поэтому не будут работать. Подготовка и обновление при прямом подключении пользователя выполняются только в момент входа пользователя.

Несколько серверов

При необходимости можно определить несколько LDAP-серверов. Например, для аутентификации разных групп пользователей можно использовать разные серверы. После настройки LDAP-серверов в конфигурации группы пользователей становится возможным выбрать нужный LDAP-сервер для соответствующей группы пользователей.

Если пользователь входит в несколько групп пользователей и для них настроено несколько LDAP-серверов, для аутентификации будет использоваться первый сервер в списке LDAP-серверов, отсортированном по имени в порядке возрастания.

Настройка

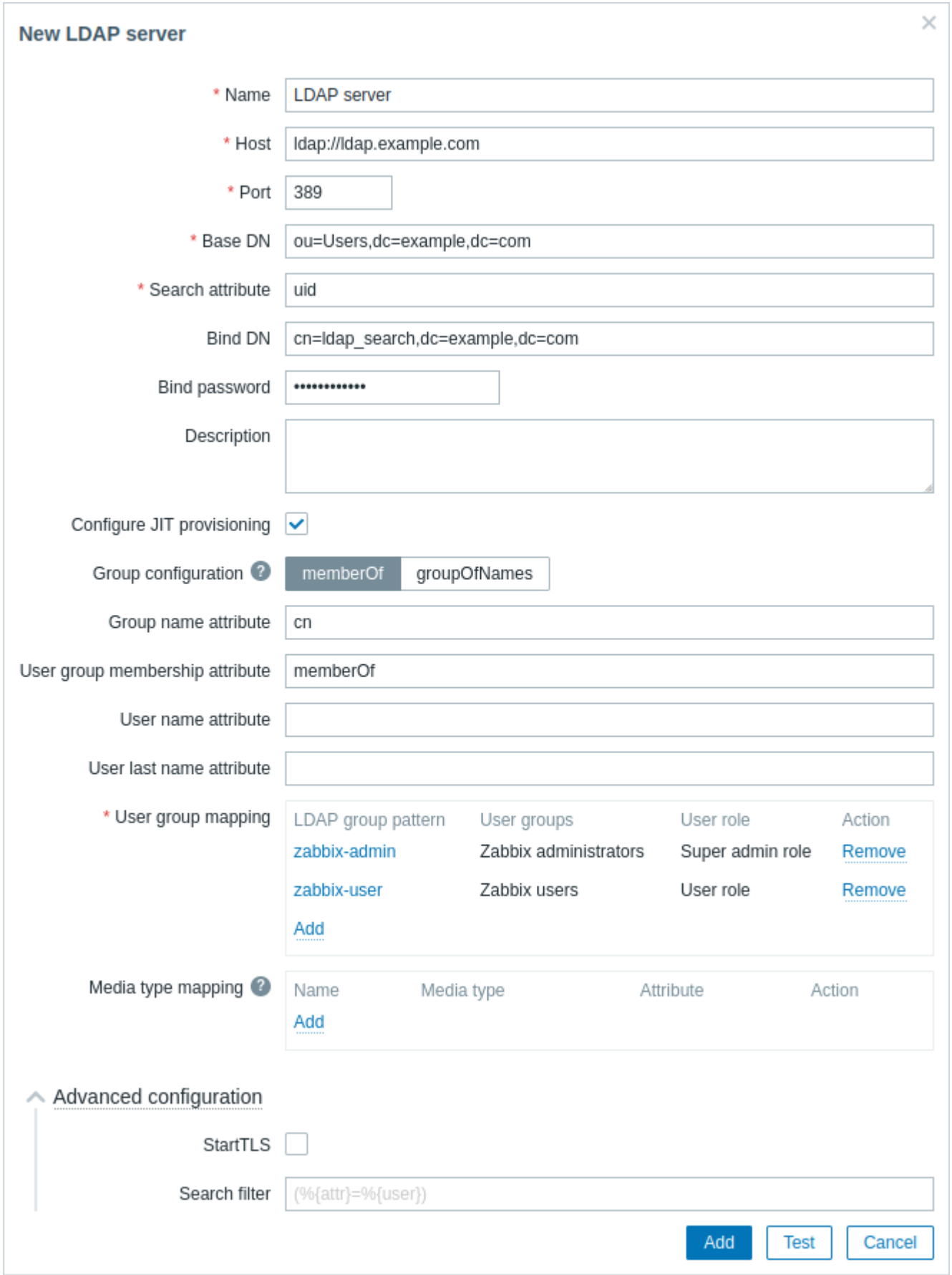


Параметры настройки:

Параметр	Описание
Включить аутентификацию LDAP	Установите флажок, чтобы включить аутентификацию LDAP.

Параметр	Описание
Включить JIT-подготовку	Установите флажок, чтобы включить JIT-подготовку.
Серверы	Нажмите *Добавить*, чтобы настроить сервер LDAP (см. раздел Настройка сервера LDAP ниже).
Учет имени пользователя с учетом регистра	Снимите флажок, чтобы отключить учет регистра при входе с использованием имен пользователей (по умолчанию включено). Отключение учета регистра при входе позволяет, например, войти как ”admin”, даже если пользователь Zabbix называется ”Admin” или ”ADMIN”. Обратите внимание: если учет регистра при входе отключен и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для этих пользователей всегда будет отклонен со следующим сообщением об ошибке: ”Authentication failed: supplied credentials are not unique.”
Период подготовки	Укажите период подготовки, то есть частоту подготовки вошедшего пользователя во время работы с веб-интерфейсом.

Конфигурация LDAP-сервера



Параметры конфигурации LDAP-сервера:

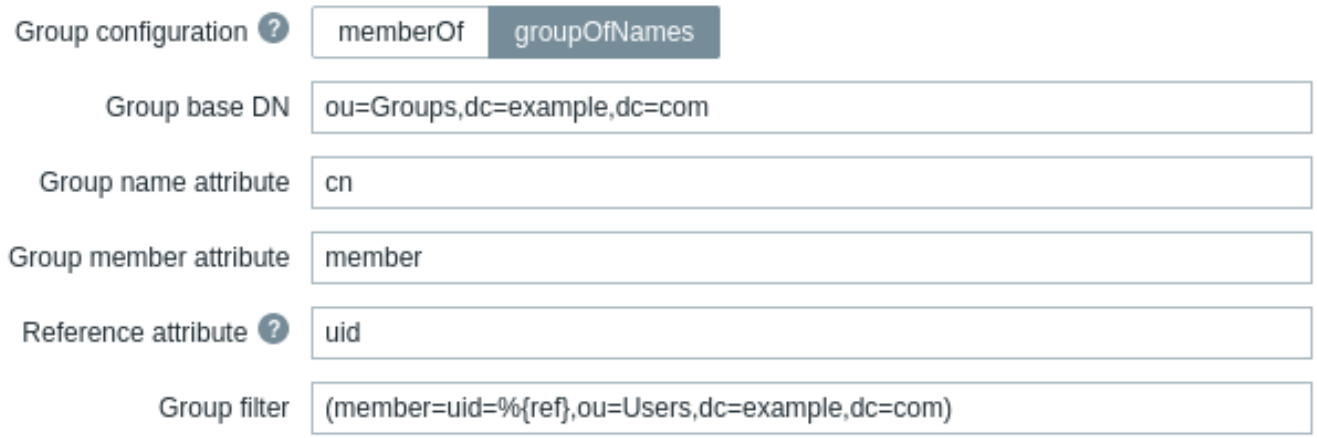
Parameter	Description
Name	Имя LDAP-сервера в конфигурации Zabbix.

Parameter	Description
Host	Имя хоста, IP-адрес или URI LDAP-сервера. Примеры: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Для защищенного LDAP-сервера используйте протокол *ldaps* и имя хоста. Пример: ldaps://ldap.example.com Для OpenLDAP 2.x.x и более поздних версий можно использовать полный LDAP URI вида ldap://hostname:port или ldaps://hostname:port.
Port	Порт LDAP-сервера. По умолчанию 389. Для защищенного LDAP-соединения номер порта обычно 636. Не используется при применении полных LDAP URI.
Base DN	Базовый путь к учетным записям пользователей на LDAP-сервере: ou=Users,ou=system (для OpenLDAP), DC=company,DC=com (для Microsoft Active Directory) uid=%{user},dc=example,dc=com (для прямой привязки пользователя, см. примечание ниже)
Search attribute	Атрибут учетной записи LDAP, используемый для поиска: uid (для OpenLDAP), sAMAccountName (для Microsoft Active Directory)
Bind DN	Учетная запись LDAP для привязки и поиска на LDAP-сервере, примеры: uid=ldap_search,ou=system (для OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (для Microsoft Active Directory) Также поддерживается анонимная привязка. Обратите внимание, что анонимная привязка потенциально открывает конфигурацию домена для неавторизованных пользователей (сведения о пользователях, компьютерах, серверах, группах, службах и т. д.). В целях безопасности отключите анонимные привязки на LDAP-хостах и используйте вместо этого аутентифицированный доступ.
Bind password	Пароль LDAP-учетной записи для привязки и поиска на LDAP-сервере.
Description	Описание LDAP-сервера.
Configure JIT provisioning	Отметьте этот флажок, чтобы показать параметры, связанные с JIT provisioning.
Group configuration	Выберите метод настройки групп: **memberOf** - путем поиска пользователей и их атрибута членства в группе **groupOfNames** - путем поиска групп через атрибут member Обратите внимание, что memberOf предпочтительнее, так как он работает быстрее; используйте groupOfNames, если ваш LDAP-сервер не поддерживает `memberOf` или требуется фильтрация групп.
Group name attribute	Укажите атрибут, из которого будет получено имя группы для всех объектов в атрибуте `memberOf` (см. поле *User group membership attribute*) Имя группы необходимо для сопоставления групп пользователей.
User group membership attribute	Укажите атрибут, содержащий сведения о группах, к которым принадлежит пользователь (например, `memberOf`). Например, атрибут memberOf может содержать такие сведения: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Это поле доступно только для метода memberOf.
User name attribute	Укажите атрибут, содержащий имя пользователя.
User last name attribute	Укажите атрибут, содержащий фамилию пользователя.
User group mapping	Сопоставьте шаблон группы пользователей LDAP с группой пользователей и ролью пользователя Zabbix. Это необходимо, чтобы определить, какую группу пользователей/роль получит создаваемый пользователь в Zabbix. Нажмите *Add*, чтобы добавить сопоставление. Поле *LDAP group pattern* поддерживает подстановочные знаки. Имя группы должно совпадать с существующей группой. Если пользователь LDAP соответствует нескольким группам пользователей Zabbix, он становится участником всех этих групп. Если пользователь соответствует нескольким ролям пользователей Zabbix, он получит роль с наивысшим уровнем прав среди них.
Media type mapping	Сопоставьте атрибуты LDAP media пользователя (например, email) с пользовательскими media Zabbix для отправки уведомлений (значение атрибута используется как поле media *Send to*).
Advanced configuration	Нажмите заголовок *Advanced configuration*, чтобы отобразить дополнительные параметры конфигурации (см. ниже).

Parameter	Description
StartTLS	Отметьте флажок, чтобы использовать операцию StartTLS при подключении к LDAP-серверу. Соединение будет прервано, если сервер не поддерживает StartTLS. StartTLS нельзя использовать с серверами, которые используют протокол *ldaps*.
Search filter	Определите пользовательскую строку при аутентификации пользователя в LDAP. Поддерживаются следующие заполнители: `%{attr}` - имя атрибута поиска (uid, sAMAccountName) `%{user}` - значение имени пользователя для аутентификации Например, чтобы выполнить поиск с учетом регистра в среде LDAP или Microsoft Active Directory, где поиск обычно не чувствителен к регистру, строку можно определить следующим образом: `(%{attr}:caseExactMatch:=%{user})`. Если фильтр не настроен, LDAP будет использовать значение по умолчанию: `(%{attr}=%{user})`.

Note:
Чтобы настроить LDAP-сервер для **прямой привязки пользователя**, добавьте атрибут uid=%{user} к параметру *Base DN* (например, *uid=%{user},dc=example,dc=com*) и оставьте параметры *BindDN* и *Bind password* пустыми. При аутентификации заполнитель %{user} будет заменен именем пользователя, введенным при входе в систему.
При прямой привязке пользователя Zabbix имеет доступ только к учетным данным пользователя во время интерактивного входа. Поэтому задачи provisioning, которые выполняются без интерактивного входа (например, кнопка *Provision now* или метод API `user.provision`), будут игнорировать пользователей, аутентифицирующихся с помощью прямой привязки пользователя, поскольку Zabbix не может выполнить привязку от их имени. Используйте анонимную привязку или отдельного пользователя для привязки (служебную учетную запись), если вам нужно, чтобы provisioning и LDAP-поиск работали из веб-интерфейса без необходимости входа конечного пользователя в систему.

Следующие поля относятся только к методу *Group configuration* = ”groupOfNames”:



Parameter	Description
Group base DN	Базовый путь к группам на LDAP-сервере.
Group name attribute	Укажите атрибут, из которого будет получено имя группы в указанном базовом пути к группам. Имя группы необходимо для сопоставления групп пользователей.
Group member attribute	Укажите атрибут, содержащий сведения об участниках группы в LDAP (например, `member`).
Reference attribute	Укажите ссылочный атрибут для фильтра группы (см. поле *Group filter*). Затем используйте `%{ref}` в фильтре группы, чтобы получить значения для указанного здесь атрибута.
Group filter	Укажите фильтр для получения группы, участником которой является пользователь. Например, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` будет соответствовать ”User1”, если атрибут member группы равен `uid=User1,ou=Users,dc=example,dc=com`, и вернет группу, участником которой является ”User1”.

Warning:
В случае проблем с сертификатами, чтобы защищенное LDAP-соединение (ldaps) заработало, может потребоваться добавить строку `TLS_REQCERT allow` в конфигурационный файл `/etc/openldap/ldap.conf`. Это может снизить безопасность соединения с LDAP-каталогом.

Note:
Рекомендуется создать отдельную LDAP-учетную запись (*Bind DN*) для выполнения привязки и поиска на LDAP-сервере с минимальными правами в LDAP вместо использования реальных учетных записей пользователей (используемых для входа в веб-интерфейс Zabbix).
 Такой подход обеспечивает более высокий уровень безопасности и не требует изменения *Bind password* при смене пользователем собственного пароля на LDAP-сервере.
 В таблице выше это учетная запись *ldap_search*.

Проверка доступа

Кнопка *Проверить* позволяет проверить доступ пользователя:

Параметр	Описание
Имя пользователя	Имя пользователя LDAP для проверки (по умолчанию указано имя текущего пользователя из веб-интерфейса Zabbix). Это имя пользователя должно существовать на сервере LDAP. Zabbix не активирует аутентификацию LDAP, если не сможет аутентифицировать тестового пользователя.
Пароль пользователя	Пароль пользователя LDAP для проверки.

SAML

Обзор

SAML 2.0 аутентификация может использоваться для входа в Zabbix.

Если вход по SAML настроен без JIT-подготовки пользователей, пользователь должен уже существовать в Zabbix; пароль пользователя Zabbix не используется для аутентификации. После успешной аутентификации SAML Zabbix сопоставляет атрибут имени пользователя, полученный из SAML, с существующим локальным именем пользователя Zabbix.

Подготовка пользователей

Можно настроить JIT (just-in-time) **подготовку пользователей** для пользователей SAML. В этом случае не требуется, чтобы пользователь уже существовал в Zabbix. Учетная запись пользователя может быть создана при первом входе пользователя в Zabbix.

Attention:
Если включена JIT-подготовка, на вкладке *Аутентификация* необходимо указать группу пользователей для деактивированных пользователей.

Помимо JIT-подготовки также можно включить и настроить SCIM (System for Cross-domain Identity Management) provisioning — *непрерывное* управление учетными записями пользователей для тех пользователей, которые были созданы с помощью подготовки пользователей. Для аутентификации в Zabbix SCIM provisioning требует API-токен Zabbix (с правами Super admin).

Например, если пользователь перемещается из одной группы SAML в другую, то в Zabbix он также будет перемещен из одной группы в другую; если пользователь удаляется из группы SAML, он также будет удален из группы в Zabbix и, если не принадлежит ни к какой другой группе, добавлен в группу пользователей для деактивированных пользователей.

Если SCIM включен и настроен, пользователь SAML будет подготовлен в момент входа пользователя в Zabbix и будет непрерывно обновляться на основе изменений в SAML. Уже существующие пользователи SAML не будут подготовлены, и обновляться будут только подготовленные пользователи. Обратите внимание, что при подготовке или обновлении пользователя будут добавляться только допустимые способы оповещения.

Если SCIM не включен, пользователь SAML будет подготовлен (а затем обновляться) в момент входа пользователя в Zabbix.

Note:
Если включена аутентификация SAML, пользователи смогут выбирать между локальным входом и единым входом через SAML. Если используется JIT-подготовка, то возможен только единый вход.

Настройка поставщика удостоверений

Чтобы работать с Zabbix, поставщик удостоверений SAML (onelogin.com, auth0.com, okta.com и т. д.) должен быть настроен следующим образом:

- *Assertion Consumer URL* должен быть установлен в `<path_to_zabbix_ui>/index_sso.php?acs`
- *Single Logout URL* должен быть установлен в `<path_to_zabbix_ui>/index_sso.php?sls`

Примеры `<path_to_zabbix_ui>`: `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_public_ip_address>/zabbix`

Настройка Zabbix

Attention:
Если вы хотите использовать аутентификацию SAML в веб-интерфейсе, необходимо установить php-openssl.

Для использования аутентификации SAML Zabbix необходимо настроить следующим образом:

1. Укажите закрытый ключ и сертификаты SP. Для активации SAML через веб-интерфейс необходимо указать сертификат IdP (вставить текст или выбрать файл). Сертификат SP и закрытый ключ SP необходимо указать, если выбраны параметры *Подписывать* или *Шифровать*.

Если в файле конфигурации веб-интерфейса (`zabbix.conf.php`) задано `$SSO['CERT_STORAGE'] = 'database'`, во время настройки SAML в веб-интерфейсе можно вставить текст сертификата или загрузить файл сертификата — файлы в файловой системе не требуются.

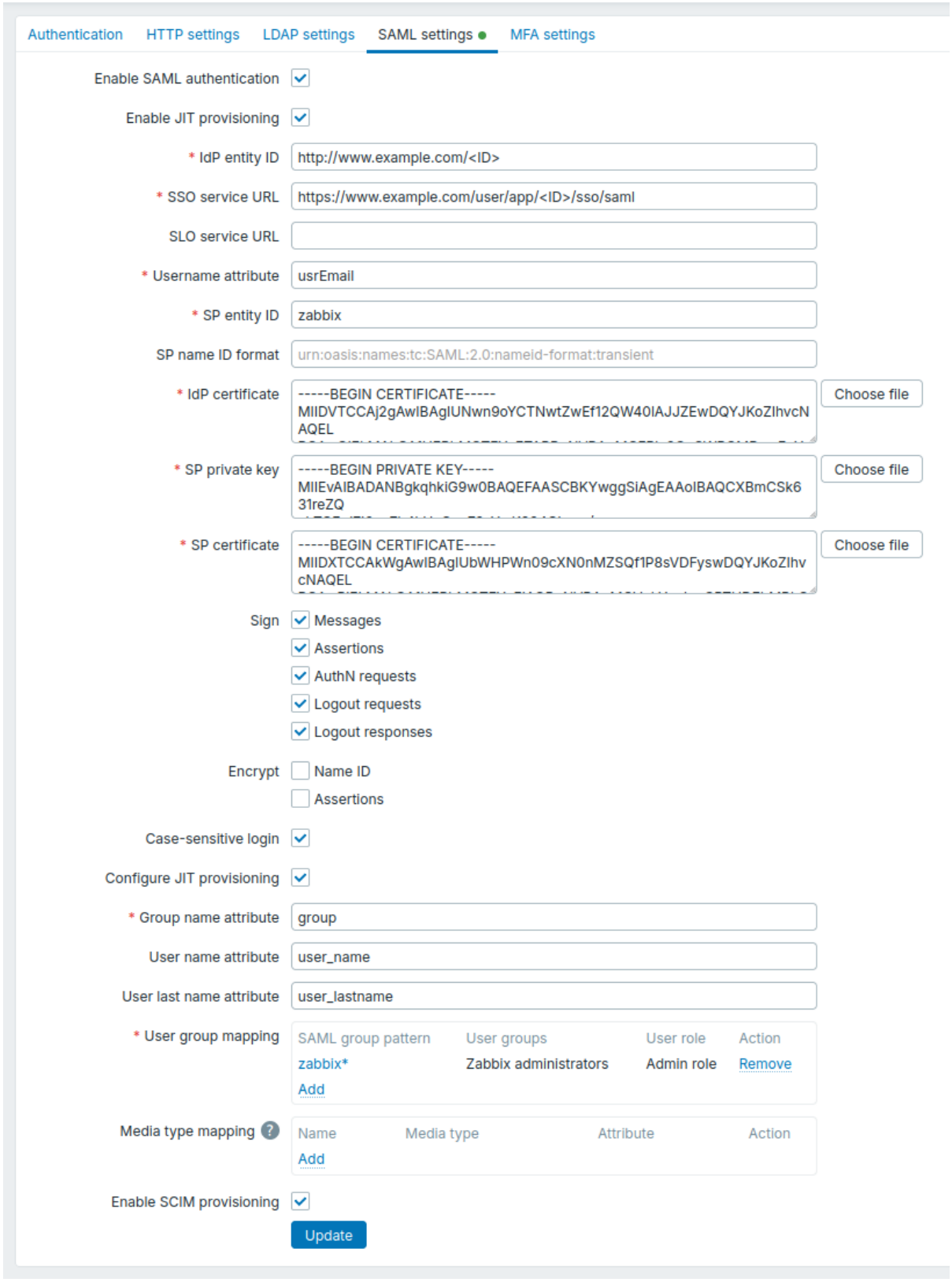
Если в файле конфигурации задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в `ui/conf/certs` или по пути, настроенному в файле конфигурации), а веб-интерфейс не будет сохранять сертификаты в базе данных. Обратите внимание: если `$SSO['CERT_STORAGE']` не задан или закомментирован, предполагается хранение в файлах.

По умолчанию Zabbix будет искать файлы в следующих расположениях:

- ui/conf/certs/sp.key - файл закрытого ключа SP
- ui/conf/certs/sp.crt - файл сертификата SP
- ui/conf/certs/idp.crt - файл сертификата IdP

Attention:
Сертификаты и ключи можно импортировать через веб-интерфейс, вставив текст или загрузив файлы. Хотя в некоторых средах для удобства допускается импорт незашифрованных данных, настоятельно рекомендуется использовать зашифрованный транспорт (например, включить HTTPS/TLS для веб-интерфейса Zabbix). Если сертификаты и ключи SAML хранятся в базе данных, включите TLS/SSL для подключений к базе данных, чтобы защитить трафик между приложением и базой данных.
 Зашифруйте резервные копии базы данных, которые могут содержать сертификаты или закрытые ключи SAML, и ограничьте доступ к файлам резервных копий и таблицам базы данных, в которых хранятся данные SAML. Ограничьте привилегии пользователя базы данных минимально необходимым уровнем.

2. Все наиболее важные параметры можно настроить в веб-интерфейсе Zabbix. Однако дополнительные параметры можно указать в файле конфигурации.



Параметры конфигурации, доступные в веб-интерфейсе Zabbix:

Параметр	Описание
Включить аутентификацию SAML	Установите флажок, чтобы включить аутентификацию SAML.
Включить JIT-подготовку	Установите флажок, чтобы включить JIT-подготовку пользователей.
Идентификатор сущности IdP	Уникальный идентификатор сущности в поставщике удостоверений SAML.
URL службы SSO	URL-адрес, на который будут перенаправляться пользователи при входе в систему.

Параметр	Описание
URL службы SLO	URL-адрес, на который будут перенаправляться пользователи при выходе из системы. Если оставить поле пустым, служба SLO использоваться не будет.
Атрибут имени пользователя	Атрибут SAML, который будет использоваться в качестве имени пользователя при входе в Zabbix. Список поддерживаемых значений определяется поставщиком удостоверений. Примеры: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
Идентификатор сущности SP	Уникальный идентификатор поставщика услуг (если он не совпадает, операция будет отклонена). Можно указать URL-адрес или любую строку данных.
Формат идентификатора имени SP	Запрос определенного формата идентификатора имени в ответе. Примеры: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified urn:oasis:names:tc:SAML:2.0:nameid-format:transient
Сертификат IdP	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа (SSO) SAML.
Закрытый ключ SP	Содержимое закрытого ключа поставщика услуг (SP) для настройки службы единого входа (SSO) SAML. Обеспечивает безопасную аутентификацию и обмен данными с поставщиком удостоверений (IdP).
Сертификат SP	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа (SSO) SAML.
Подписывать	Снимите флажки, чтобы отключить подписи SAML для соответствующих сущностей: *Сообщения* *Утверждения* *Запросы AuthN* *Запросы выхода из системы* *Ответы на выход из системы*
Шифровать	Установите флажки, чтобы выбрать сущности, для которых необходимо включить шифрование SAML: *Идентификатор имени* *Утверждения*
Учет имени пользователя с учетом регистра	Снимите флажок, чтобы отключить учет регистра при входе пользователей (по умолчанию включен). Отключение учета регистра позволяет, например, войти как ”admin”, даже если пользователь Zabbix называется ”Admin” или ”ADMIN”. Обратите внимание: если учет регистра отключен и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для этих пользователей всегда будет запрещен со следующим сообщением об ошибке: ”Authentication failed: supplied credentials are not unique.”
Настроить JIT-подготовку	Установите этот флажок, чтобы отобразить параметры, связанные с JIT-подготовкой пользователей.
Атрибут имени группы	Укажите атрибут имени группы для JIT-подготовки пользователей.
Атрибут имени пользователя	Укажите атрибут имени пользователя для JIT-подготовки пользователей.
Атрибут фамилии пользователя	Укажите атрибут фамилии пользователя для JIT-подготовки пользователей.

Параметр	Описание
Сопоставление групп пользователей	Сопоставьте шаблон группы пользователей SAML с группой пользователей и ролью пользователя Zabbix. Это необходимо, чтобы определить, какую группу и роль получит подготовленный пользователь в Zabbix. Нажмите *Добавить*, чтобы добавить сопоставление. В поле *Шаблон группы SAML* поддерживаются подстановочные знаки. Имя группы должно соответствовать существующей группе. Если пользователь SAML соответствует нескольким группам пользователей Zabbix, он становится участником всех этих групп. Если пользователь соответствует нескольким ролям пользователей Zabbix, он получит наивысший уровень разрешений среди них.
Сопоставление типов носителей	Сопоставьте атрибуты носителей пользователя SAML (например, адрес электронной почты) с носителями пользователя Zabbix для отправки уведомлений.
Включить подготовку SCIM	Установите этот флажок, чтобы включить подготовку SCIM 2.0.

Примеры настройки поставщиков удостоверений SAML для входа и подготовки пользователей в Zabbix:

- Microsoft Azure AD
- Okta
- Onelogin

Примечания о подготовке SCIM

Для подготовки SCIM укажите путь к веб-интерфейсу Zabbix и добавьте к нему `api_scim.php` на стороне поставщика удостоверений, то есть:

`https://<path-to-zabbix-ui>/api_scim.php`

Атрибуты пользователей, используемые в Zabbix (имя пользователя, имя, фамилия пользователя и атрибуты средств связи), необходимо добавить как пользовательские атрибуты. При необходимости внешнее пространство имён должно совпадать со схемой пользователя: `urn:ietf:params:scim:schemas:core:2.0:User`.

Расширенные настройки

Дополнительные параметры SAML можно настроить в файле конфигурации веб-интерфейса Zabbix (`zabbix.conf.php`):

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Примеры конфигурации

Пример 1. Подключение к экземпляру Zabbix с использованием балансировщика нагрузки через TLS/SSL

Если для подключения к экземпляру Zabbix используется балансировщик нагрузки, который применяет TLS/SSL, а Zabbix его не использует, укажите параметры `baseurl`, `strict` и `use_proxy_headers` следующим образом:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Пример 2. Настройка пользовательских алгоритмов подписи и дайджеста SAML

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Пример 3. Настройка единого входа (SSO) с аутентификацией Kerberos с использованием ADFS

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

В этом случае в поле *SP name ID* конфигурации SAML укажите:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

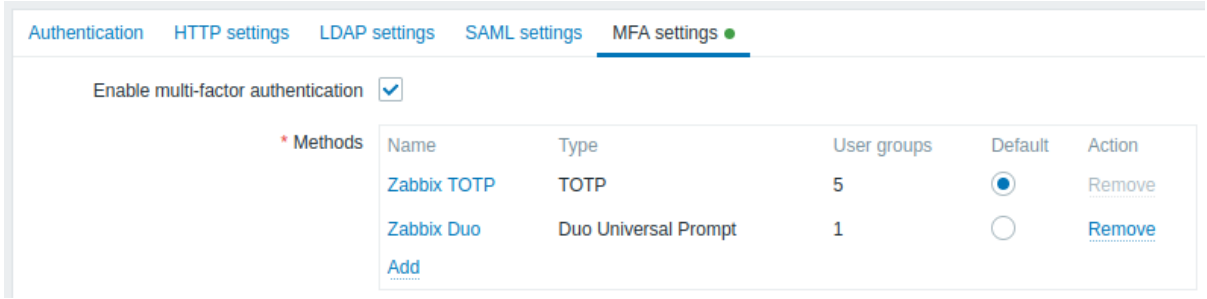
Обзор

Многофакторная аутентификация (MFA) может использоваться для входа в Zabbix, обеспечивая дополнительный уровень безопасности помимо одного только имени пользователя и пароля.

При использовании MFA пользователь должен существовать в Zabbix, должен указывать учетные данные Zabbix при входе в систему, а также должен подтвердить свою личность другим способом, обычно с помощью кода, сгенерированного приложением-аутентификатором на телефоне пользователя.

Доступно несколько методов MFA, что позволяет пользователям выбрать вариант, который лучше всего соответствует их требованиям к безопасности и предпочтениям. Эти методы: одноразовый пароль на основе времени (TOTP) и Duo Universal Prompt.

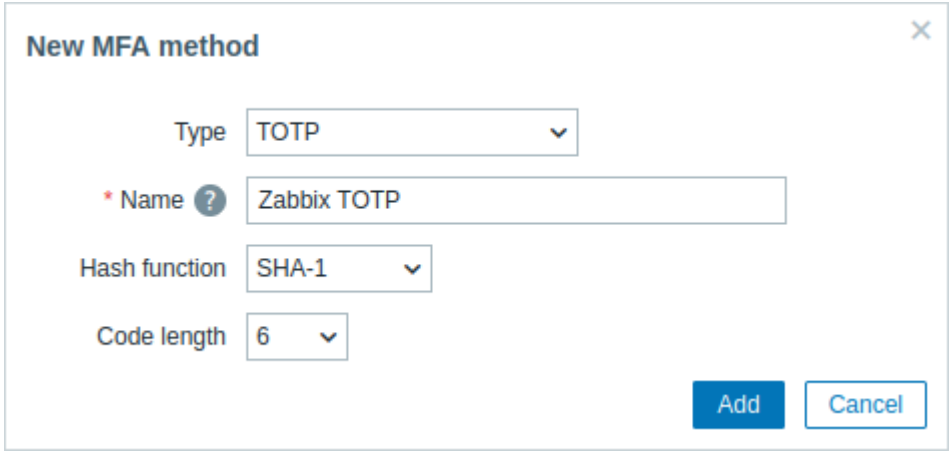
Конфигурация



Параметры конфигурации:

Parameter	Description
Enable multi-factor authentication	Установите флажок, чтобы включить многофакторную аутентификацию.
Methods	Нажмите *Add*, чтобы настроить метод MFA (см. конфигурация метода ниже).

Настройка метода



Параметры настройки метода:

Параметр	Описание
Тип	Выберите тип метода MFA: **TOTP** - используйте приложение-аутентификатор для генерации одноразовых паролей на основе времени; **Duo Universal Prompt** - используйте службу аутентификации Duo для обеспечения многофакторной аутентификации.
Имя	Введите имя, которое будет отображаться как имя учетной записи для всех пользователей MFA в приложениях-аутентификаторах (например, ”Zabbix”).
Функция хеширования	Выберите функцию хеширования (SHA-1, SHA-256 или SHA-512) для генерации кодов TOTP. Этот параметр доступен, если тип метода MFA установлен в ”TOTP”. Обратите внимание, что выбор SHA-256 или SHA-512 может значительно ограничить совместимость, поскольку многие приложения в настоящее время не поддерживают эти функции.
Длина кода	Выберите длину кода подтверждения (6 или 8). Этот параметр доступен, если тип метода MFA установлен в ”TOTP”.
Имя хоста API	Введите имя хоста API, предоставленное службой аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в ”Duo Universal Prompt”.
ID клиента	Введите ID клиента, предоставленный службой аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в ”Duo Universal Prompt”.
Секрет клиента	Введите секрет клиента, предоставленный службой аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в ”Duo Universal Prompt”.

Примеры конфигурации

В этом разделе приведены примеры настройки MFA с использованием Time-Based One-Time Password (TOTP) и Duo Universal Prompt.

TOTP

Для TOTP пользователи должны подтвердить свою личность с помощью приложения-аутентификатора (например, приложения Google Authenticator).

1. Перейдите в настройки MFA в Zabbix в разделе *Users* > *Authentication* и включите многофакторную аутентификацию.

2. Добавьте новый MFA метод со следующей конфигурацией:

- Type: TOTP
- Name: Zabbix TOTP
- Hash function: SHA-1
- Code length: 6

3. Нажмите кнопку *Add*, а затем кнопку *Update*.

4. Перейдите в *Users* > *User groups* и создайте новую группу пользователей со следующей конфигурацией:

- Group name: TOTP group
- Users: Admin
- Multi-factor authentication: Default (или ”Zabbix TOTP”, если это не задано как значение по умолчанию)

5. Выйдите из Zabbix и войдите снова, используя свои учетные данные. После успешного входа вам будет предложено зарегистрироваться в MFA; при этом отобразятся QR-код и секретный ключ.



6. Отсканируйте QR-код или введите секретный ключ в приложение Google Authenticator. Приложение сгенерирует код подтверждения, который нужно ввести для завершения процесса входа.

7. При последующих входах получите код подтверждения в приложении Google Authenticator и введите его при входе.

Duo Universal Prompt

Для Duo Universal Prompt пользователи должны подтверждать свою личность с помощью приложения-аутентификатора Duo Mobile.

Attention:
Метод MFA Duo Universal Prompt требует установки расширения php-curl, доступа к Zabbix по HTTPS и разрешения на исходящие подключения к серверам Duo. Кроме того, если вы включили Content Security Policy (CSP) на веб-сервере, обязательно добавьте "duo.com" в директиву CSP в файле конфигурации вашего виртуального хоста.

1. Зарегистрируйте бесплатную учетную запись администратора Duo на Duo Signup.

2. Откройте панель Duo Admin Panel, перейдите в *Applications* > *Protect an Application*, найдите приложение *Web SDK* и нажмите *Protect*.

3. Запишите учетные данные (Client ID, Client secret, API hostname), необходимые для настройки метода MFA в Zabbix.

4. Перейдите в настройки MFA в Zabbix в разделе *Users* > *Authentication* и включите многофакторную аутентификацию.

5. Добавьте новый метод MFA со следующей конфигурацией:

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (use API hostname from Duo)
- Client ID: (use Client ID from Duo)
- Client secret: (use Client secret from Duo)

6. Нажмите кнопку *Add*, а затем кнопку *Update*.

7. Перейдите в *Users* > *User groups* и создайте новую группу пользователей со следующей конфигурацией:

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default (or "Zabbix Duo" if it is not set as default)

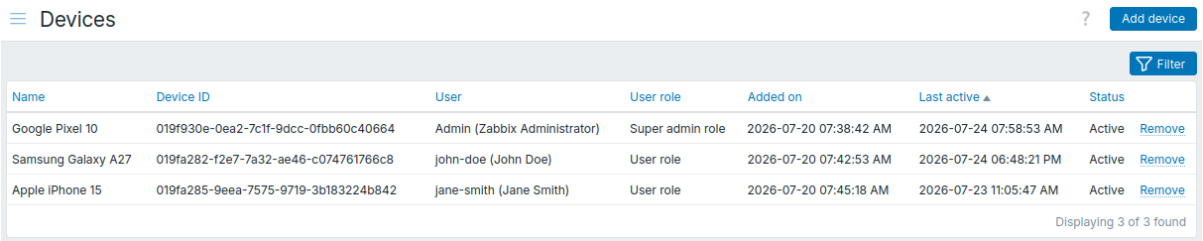
8. Выйдите из Zabbix и войдите снова, используя свои учетные данные. После успешного входа вам будет предложено зарегистрироваться в MFA, после чего вы будете перенаправлены в Duo. Завершите настройку Duo и подтвердите свою учетную запись с помощью приложения Duo на телефоне, чтобы войти в систему.

9. Для последующих входов используйте соответствующий метод MFA, предоставляемый приложением Duo (например, получение кода подтверждения, ответ на push-уведомления или использование аппаратных ключей), и введите требуемую информацию во время входа.

Устройства

В этом разделе суперадминистраторы могут управлять мобильными устройствами, которые будут использоваться с мобильным приложением Zabbix.

Этот раздел отображается, если включен параметр `EnableMobileDevices`.



Чтобы добавить устройство:

1. Нажмите *Добавить устройство*.
2. Выберите пользователя и нажмите *Добавить*. У пользователя должен быть включен доступ к устройствам.
3. Появится QR-код. Пользователь должен отсканировать этот код в мобильном приложении Zabbix.

Note:
При добавлении устройства для пользователя Zabbix также добавляет тип носителя push-уведомления в носители пользователя. При удалении устройства этот носитель пользователя также удаляется. Если настроено несколько типов носителей для push-уведомлений, Zabbix добавляет включенный тип, который первым идет в алфавитном порядке.

Добавленные устройства отображаются в списке, который можно фильтровать.

Также можно удалять устройства. При удалении устройства его учетные данные доступа отзываются и удаляются из Zabbix, после чего оно больше не сможет получить доступ к Zabbix.

В списке отображается следующая информация:

Столбец	Описание
Имя	Имя устройства.
ID устройства	ID устройства.
Пользователь	Пользователь, для которого было добавлено устройство, или *Недоступный пользователь*, если этот пользователь был удален.
Роль пользователя	Роль пользователя.
Добавлено	Дата и время добавления устройства.
Последняя активность	Дата и время последней активности устройства.
Состояние	Состояние устройства: **Новое** - устройство было добавлено без сканирования QR-кода, его потребуется добавить снова. **Новое (срок действия токена истек)** - устройство было добавлено без сканирования QR-кода, срок действия токена в QR-коде истек; устройство потребуется добавить снова. **Активно** - устройство было добавлено, QR-код отсканирован, и устройство может использовать мобильное приложение Zabbix. **Не назначено** - устройство было добавлено, но его пользователь был удален, поэтому устройство не может использовать мобильное приложение Zabbix.

Пользователи с достаточными правами также могут управлять собственными устройствами в разделе *Настройки пользователя* > *Устройства*.

Администрирование

Обзор

Меню Администрирование используется в Zabbix для административных функций. Это меню доступно только пользователям с типом Супер-администратор.

Источник данных

Обзор

В разделе *Администрирование* > *Источник данных* можно указать источник данных телеметрии, то есть базу данных, в которой хранятся данные телеметрии (трассировки, метрики и журналы). Данные телеметрии используются для мониторинга производительности приложений (APM), который помогает отслеживать прохождение запросов через приложения и службы, измерять их производительность и находить причины медленных или неудачных операций.

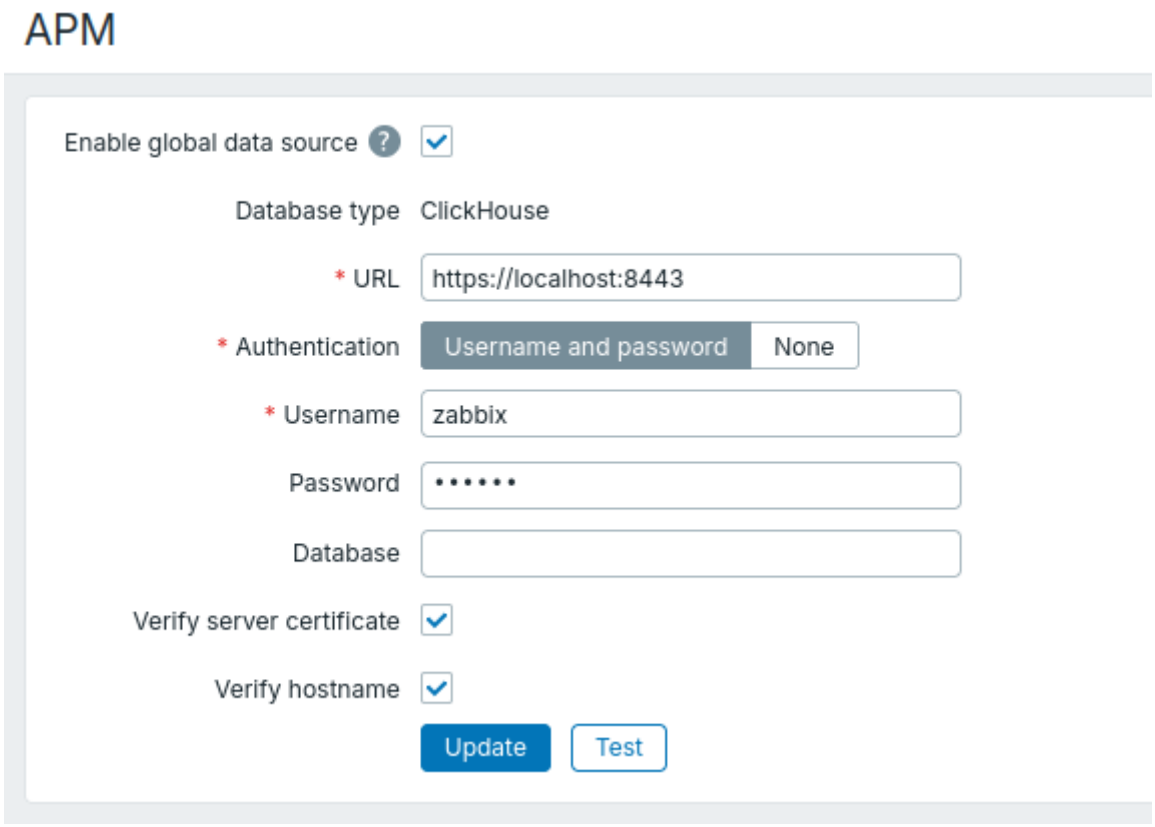
Источник данных телеметрии должен поддерживать OpenTelemetry.

Источник данных телеметрии, указанный в этом разделе, называется глобальным источником данных, поскольку он используется сервером Zabbix, прокси и веб-интерфейсом. Однако его можно переопределить для любого из этих компонентов, указав другой источник данных телеметрии в конфигурации сервера, прокси или веб-интерфейса.

Элемент данных типа запрос телеметрии запрашивает данные из этого источника, если в конфигурации компонента не указан другой источник.

APM

Чтобы указать глобальный источник данных, установите флажок *Включить глобальный источник данных* и заполните описанные ниже параметры. Перед сохранением можно нажать *Тест*, чтобы проверить подключение к этому источнику данных (требуется расширение php-curl).



Параметр	Описание
Тип базы данных	Тип базы данных жестко задан как *ClickHouse* и не может быть изменен. См. также: Требования и Настройка ClickHouse.
URL	Введите URL HTTP или HTTPS, используемый для подключения к базе данных.
Аутентификация	Выберите метод аутентификации для входа в базу данных: *Имя пользователя и пароль* или *Нет*.
Имя пользователя	Укажите имя пользователя, используемое для входа в базу данных. Этот параметр доступен, если для параметра *Аутентификация* установлено значение *Имя пользователя и пароль*.
Пароль	Укажите пароль, используемый для входа в базу данных, или оставьте поле пустым, если у пользователя базы данных нет пароля. Пароль сбрасывается при изменении URL. Этот параметр доступен, если для параметра *Аутентификация* установлено значение *Имя пользователя и пароль*.
База данных	Введите имя базы данных, содержащей данные телеметрии. Если оставить поле пустым, Zabbix использует базу данных по умолчанию, настроенную в ClickHouse для этого пользователя.
Проверять сертификат сервера	Установите флажок, чтобы проверять сертификат сервера. Если флажок установлен, соответствующие параметры TLS должны быть заданы в конфигурации сервера, прокси или веб-интерфейса. Этот параметр доступен, если *URL* содержит URL HTTPS.
Проверять имя узла сети	Установите флажок, чтобы проверить соответствие имени узла сети в *URL* имени узла сети в сертификате сервера. Если флажок установлен, соответствующие параметры TLS должны быть заданы в конфигурации сервера, прокси или веб-интерфейса. Этот параметр доступен, если *URL* содержит URL HTTPS.

Общие сведения

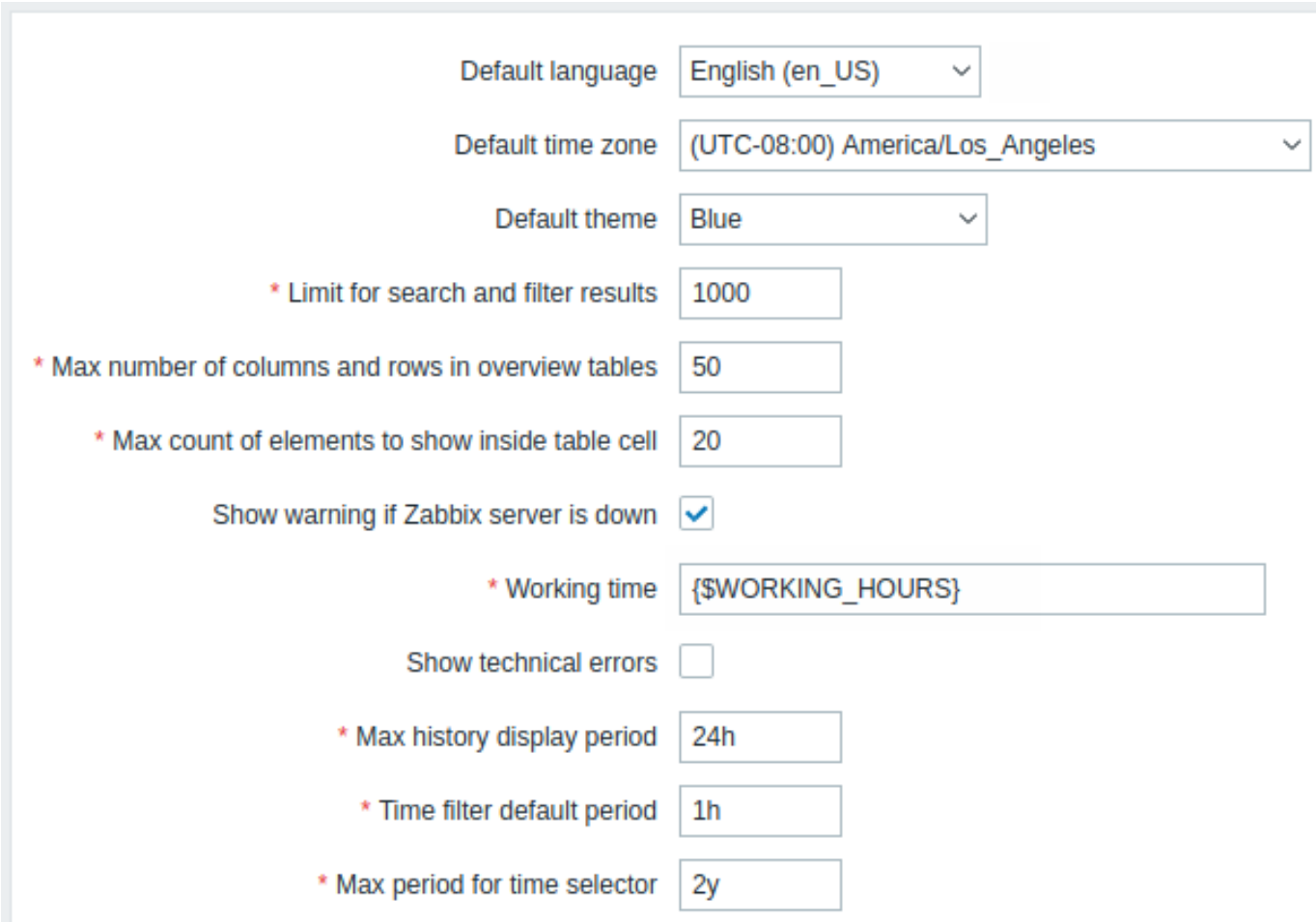
Обзор

Раздел *Administration* > *General* содержит ряд подразделов для настройки значений по умолчанию, связанных с веб-интерфейсом, и для настройки Zabbix.

Список доступных подразделов отображается при нажатии *General* в разделе меню *Administration*. Также можно переключаться между подразделами с помощью выпадающего списка заголовка в левом верхнем углу.

GUI

В этом разделе настраиваются несколько значений по умолчанию, связанных с веб-интерфейсом.



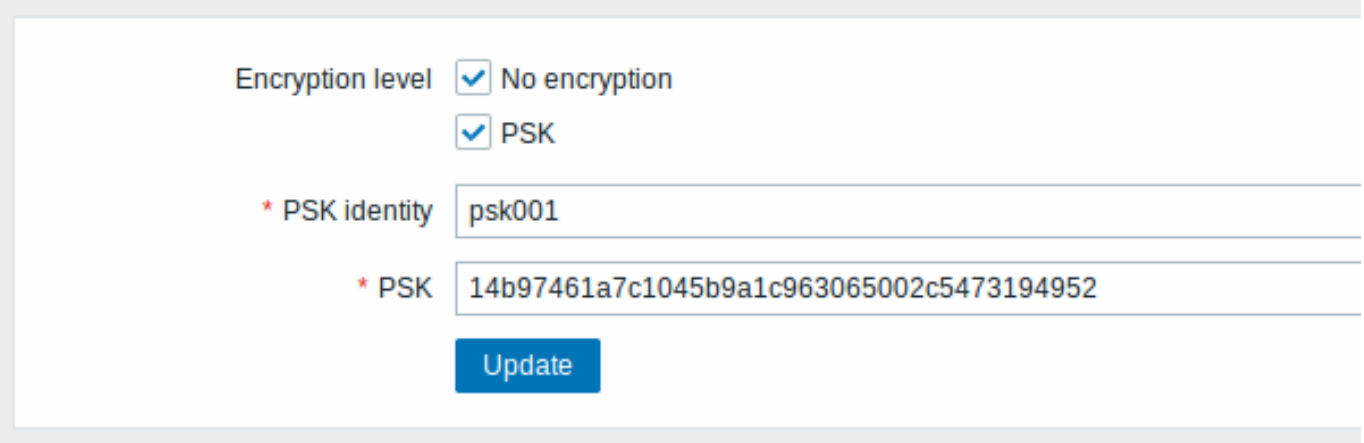
Параметры конфигурации:

Parameter	Description
Default language	Язык по умолчанию для пользователей, которые не указали язык в своих профилях, и для гостевых пользователей. Дополнительные сведения см. в разделе Установка дополнительных языков веб-интерфейса.
Default time zone	Часовой пояс по умолчанию для пользователей, которые не указали часовой пояс в своих профилях, и для гостевых пользователей.
Default theme	Тема по умолчанию для пользователей, которые не указали тему в своих профилях, и для гостевых пользователей.
Limit for search and filter results	Максимальное количество элементов (строк), которое будет отображаться в списке веб-интерфейса, например, в *Сбор данных* > *Узлы сети*. *Примечание*: если задать, например, значение ’50’, во всех затронутых списках веб-интерфейса будут отображаться только первые 50 элементов. Если в каком-либо списке содержится более пятидесяти элементов, это будет обозначено знаком ’+’ в сообщении *”Отображаются элементы с 1 по 50 из найденных **50+**”*. Кроме того, если используется фильтрация и совпадений все равно больше 50, будут отображены только первые 50. Обратите внимание, что увеличение значения этого параметра может привести к снижению производительности и увеличению потребления памяти на стороне веб-сервера.
Max number of columns and rows in overview tables	Максимальное количество столбцов и строк, отображаемых в виджете панели *Обзор триггеров*. Этот же предел применяется и к столбцам, и к строкам. Если существует больше строк и/или столбцов, чем показано, внизу таблицы будет отображаться предупреждение: ”Не все результаты отображены. Уточните критерии поиска.”
Max count of elements to show inside table cell	Для записей, отображаемых в одной ячейке таблицы, будет показано не больше указанного здесь количества.

Parameter	Description
Show warning if Zabbix server is down	Этот параметр включает отображение предупреждающего сообщения в окне браузера, если сервер Zabbix недоступен (возможно, не работает). Сообщение остается видимым даже при прокрутке страницы вниз. При наведении курсора сообщение временно скрывается, чтобы показать содержимое под ним.
Working time	Этот общесистемный параметр определяет рабочие часы. На графиках рабочее время отображается белым фоном, а нерабочее - серым. Описание формата времени см. на странице Спецификация временных периодов. Поддерживаются пользовательские макросы.
Show technical errors	Если этот флажок установлен, все зарегистрированные пользователи смогут видеть технические ошибки (PHP/SQL). Если флажок снят, информация доступна только суперадминистраторам Zabbix и пользователям, входящим в группы пользователей с включенным режимом отладки.
Max history display period	Максимальный период времени, за который отображаются исторические данные в *Мониторинг* > *Последние данные*, информация о веб-сценарии узла сети в *Мониторинг* > *Узлы сети* и в виджете панели *Топ элементов данных*. Допустимый диапазон: 24 часа (по умолчанию) - 1 неделя. Поддерживаются суффиксы времени, например 1w (одна неделя), 36h (36 часов).
Time filter default period	Период времени, используемый по умолчанию в графиках и панелях. Допустимый диапазон: 1 минута - 10 лет (по умолчанию: 1 час). Поддерживаются суффиксы времени, например 10m (десять минут), 5w (пять недель). Примечание: когда пользователь изменяет период времени при просмотре графика, этот период сохраняется как пользовательское предпочтение, заменяя глобальное значение по умолчанию или предыдущий выбор пользователя.
Max period for time selector	Максимально доступный период времени для графиков и панелей. Пользователи не смогут просматривать более старые данные. Допустимый диапазон: 1 год - 10 лет (по умолчанию: 2 года). Поддерживаются суффиксы времени, например 1y (один год), 365w (365 недель).

Авторегистрация

На этом экране вы можете настроить уровень шифрования для автоматической регистрации активного агента.



Параметры, отмеченные звёздочкой, являются обязательными.

Параметры настроек:

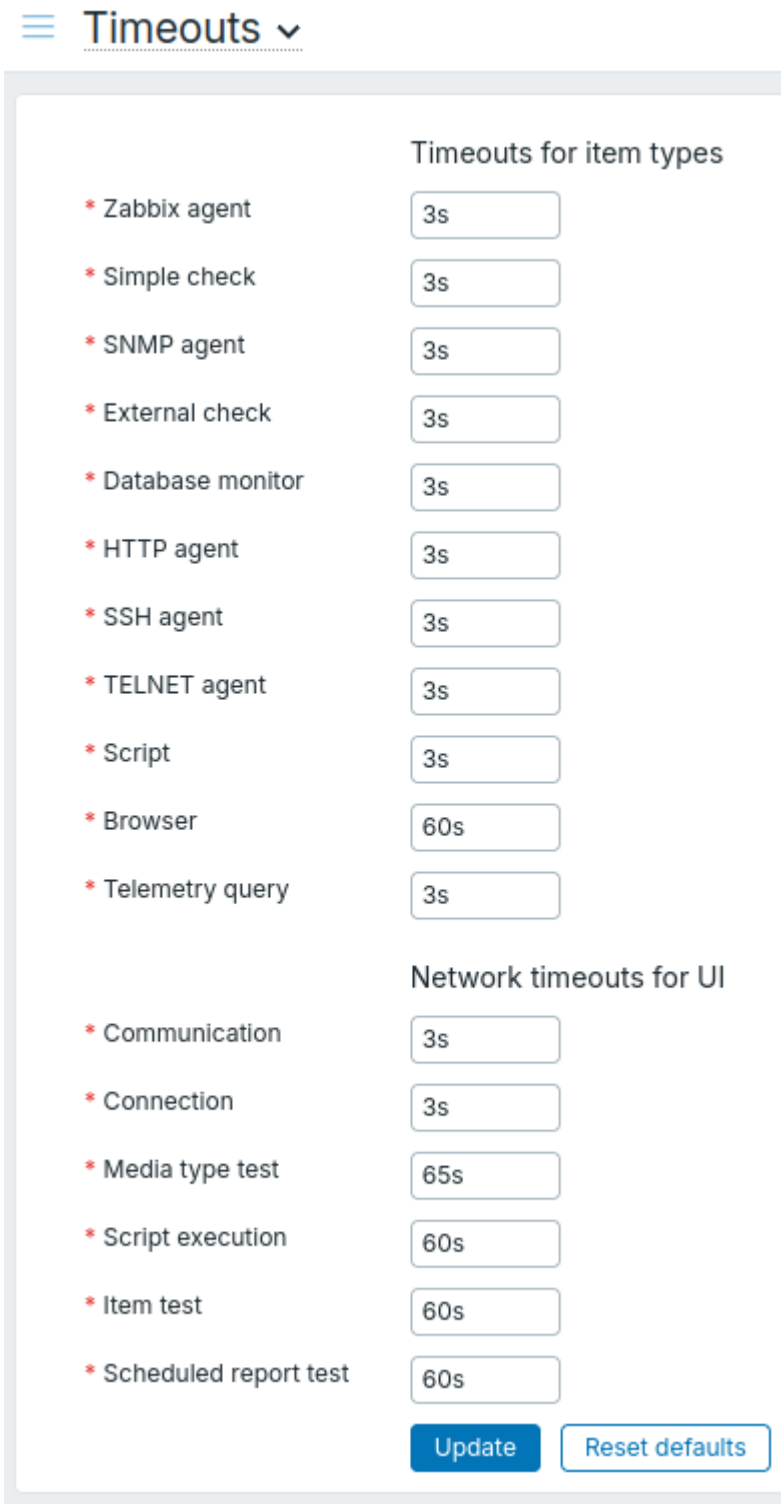
Параметр	Описание
Уровень шифрования (Encryption level)	Выберите один или оба параметра для уровня шифрования: **Без шифрования (No encryption)** — разрешены незашифрованные соединения **PSK** — разрешены зашифрованные соединения TLS с общим ключом (pre-shared key)
Идентификатор PSK (PSK identity)	Введите строку идентификатора общего ключа. Это поле доступно только в том случае, если для параметра *Уровень шифрования (Encryption level)* выбрано «PSK». Не помещайте конфиденциальную информацию в идентификатор PSK, она передается по сети незашифрованной для информирования получателя, какой PSK использовать.

Параметр	Описание
PSK	Введите общий ключ (чётное число шестнадцатеричных символов). Максимальная длина: 512 шестнадцатеричных цифр (256-байтовый PSK), если Zabbix использует библиотеку GnuTLS или OpenSSL, 64 шестнадцатеричных цифры (32-байтовый PSK), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Это поле доступно только в том случае, если для параметра *Уровень шифрования (Encryption level)* выбрано «PSK».

Смотрите также: Безопасная авторегистрация

Таймауты

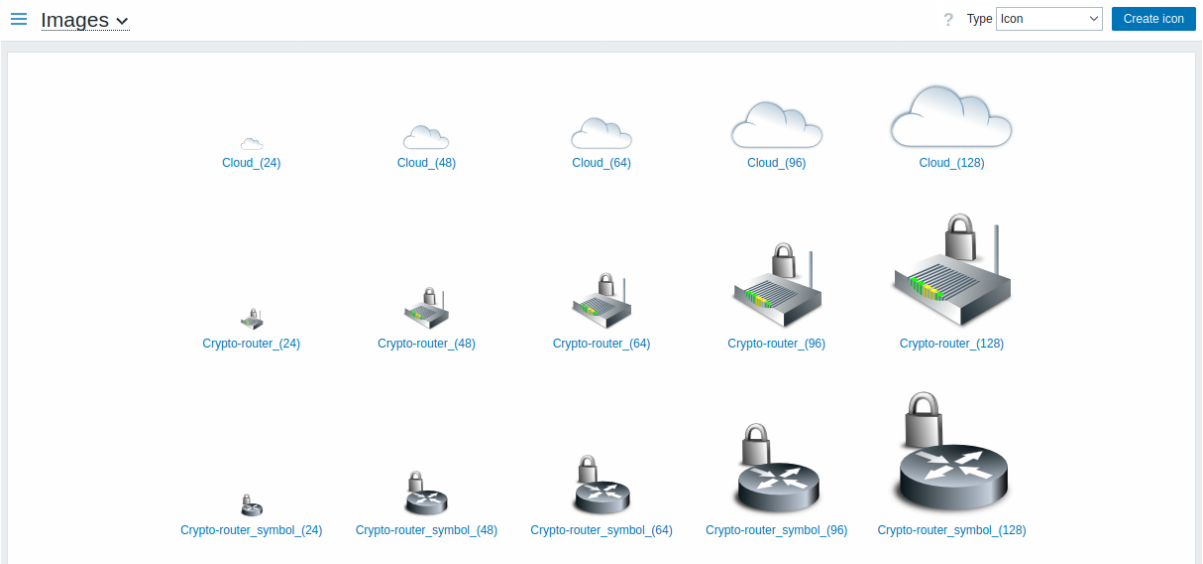
В этом разделе можно задать глобальные таймауты для типов элементов данных и сетевые таймауты. Все поля в этой форме обязательны.



Параметр		Описание
Таймауты для типов элементов данных		Сколько секунд Zabbix должен ждать перед прерыванием проверки (в зависимости от ее типа). Дополнительные сведения и ограничения см. в разделе таймаут элемента данных. Допустимый диапазон: 1 - 600 с (по умолчанию: 3 с; по умолчанию для элементов данных типа browser: 60 с). Поддерживаются суффиксы времени, например 30s, 1m, а также пользовательские макросы. Обратите внимание, что таймауты на уровне прокси, если они настроены, переопределяют эти глобальные параметры. Кроме того, если настроены таймауты для конкретных элементов данных, они переопределяют как глобальные параметры, так и параметры прокси.
Сетевые таймауты для веб-интерфейса		
	Обмен данными	Сколько секунд ждать перед закрытием неактивного сокета (если соединение с сервером Zabbix уже было установлено ранее, но веб-интерфейс не может завершить операцию чтения/отправки данных в течение этого времени, соединение будет разорвано). Допустимый диапазон: 1 - 300 с (по умолчанию: 3 с).
	Соединение	Сколько секунд ждать перед прекращением попытки подключения к серверу Zabbix. Допустимый диапазон: 1 - 30 с (по умолчанию: 3 с).
	Тест типа носителя	Сколько секунд ждать ответа при тестировании типа носителя. Допустимый диапазон: 1 - 300 с (по умолчанию: 65 с).
	Выполнение скрипта	Сколько секунд ждать ответа при выполнении скрипта. Допустимый диапазон: 1 - 300 с (по умолчанию: 60 с). Этот таймаут действует на всю цепочку выполнения скрипта, длина которой может быть различной. Например, если скрипт выполняется на агенте, это означает проход через сервер (возможно, также через прокси) к агенту и обратно. Для скриптов, выполняемых на агенте, применяется таймаут сервер или прокси. Для скриптов, выполняемых только на активном агенте, вероятно, потребуется увеличить значение таймаута по умолчанию для сервера/прокси. Таймаут сервера/прокси должен быть больше частоты обновления активной проверки, иначе таймаут сервера/прокси может быть превышен еще до того, как активный агент получит скрипт. См. также: Таймаут скрипта.
	Тест элемента данных	Сколько секунд ждать возвращаемых данных при тестировании элемента данных. Допустимый диапазон: 1 - 600 с (по умолчанию: 60 с).
	Тест запланированного отчета	Сколько секунд ждать возвращаемых данных при тестировании запланированного отчета. Допустимый диапазон: 1 - 300 с (по умолчанию: 60 с).

Изображения

В разделе «Изображения» отображаются все изображения, доступные в Zabbix.
Изображения хранятся в базе данных.

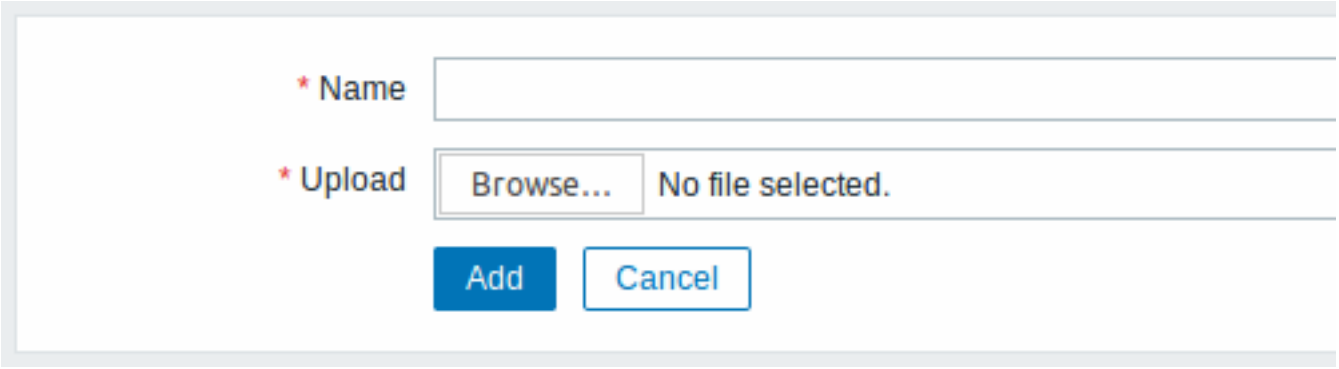


С помощью раскрывающегося списка *Тип* можно переключаться между значками и фоновыми изображениями:

- Значки используются для отображения элементов сетевой карты.
- Фоновые изображения используются в качестве фона сетевых карт.

Добавление изображения

Чтобы добавить собственное изображение, нажмите кнопку *Создать значок* или *Создать фон* в правом верхнем углу.



Атрибуты изображения:

Параметр	Описание
Имя	Уникальное имя изображения.
Загрузка	Выберите файл (PNG, JPEG, GIF или WebP) в локальной системе для загрузки в Zabbix. *Примечание*: может быть возможна загрузка файлов других форматов, которые во время загрузки будут преобразованы в PNG. Для обработки изображений используется библиотека GD, поэтому поддерживаемые форматы зависят от используемой версии библиотеки (Zabbix требуется версия 2.0.28 или выше).

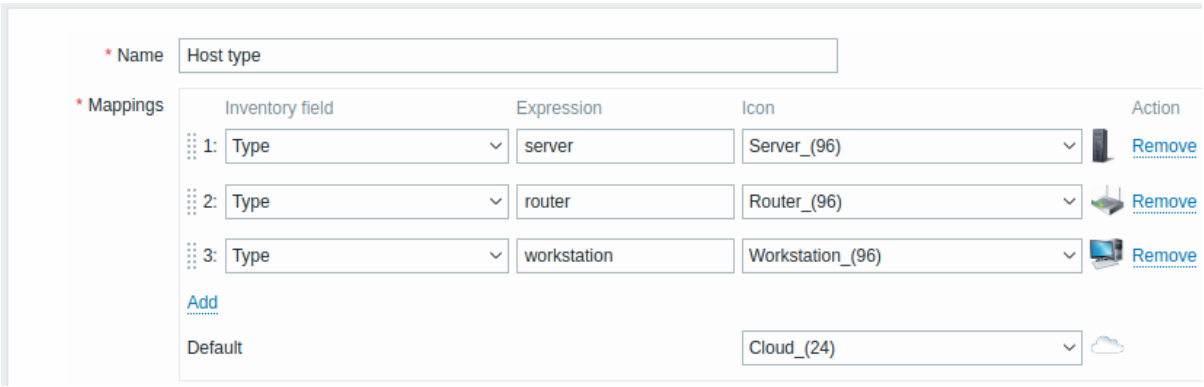
Note:
Максимальный размер загружаемого файла ограничен значением `ZBX_MAX_IMAGE_SIZE`, которое составляет 1024x1024 байта или 1 МБ.
Загрузка изображения может завершиться ошибкой, если его размер близок к 1 МБ, а параметр конфигурации MySQL `max_allowed_packet` имеет значение по умолчанию 1 МБ.
В этом случае увеличьте значение параметра `max_allowed_packet`.

Сопоставление значков

Этот раздел позволяет создавать сопоставление определенных узлов сети с определенными значками. Для создания сопоставления используется информация из полей инвентаризации узла сети.

Затем эти сопоставления можно использовать в настройке карты сети, чтобы автоматически назначать соответствующие значки подходящим узлам сети.

Чтобы создать новую карту значков, нажмите *Create icon map* в правом верхнем углу.



Параметры конфигурации:

Параметр	Описание
Name	Уникальное имя карты значков.
Mappings	Список сопоставлений. Порядок сопоставлений определяет, какое из них будет иметь приоритет. Вы можете перемещать сопоставления вверх и вниз по списку с помощью перетаскивания.
Inventory field	Поле инвентаризации узла сети, которое будет проверяться на совпадение.
Expression	Регулярное выражение, описывающее совпадение.
Icon	Значок, который будет использоваться, если найдено совпадение с выражением.
Default	Значок, используемый по умолчанию.

Регулярные выражения

Этот раздел позволяет создавать пользовательские регулярные выражения, которые можно использовать в нескольких местах веб-интерфейса. Подробности см. в разделе Регулярные выражения.

Параметры отображения триггеров

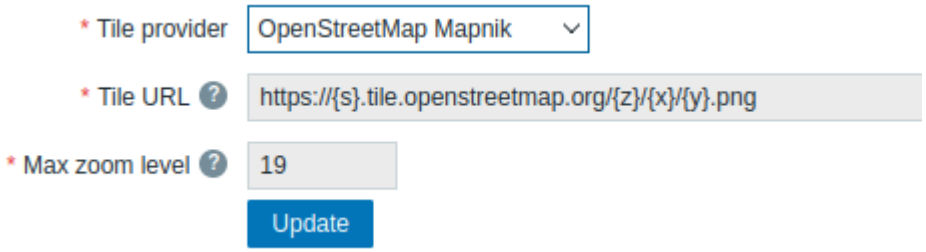
Этот раздел позволяет настраивать, как состояние триггера отображается на веб-интерфейсе, а также имена и цвета серьезности триггера.



Parameter	Description
Использовать пользовательские цвета статуса событий	При включении этого параметра настраиваются цвета для подтвержденных/неподтвержденных проблем.
Неподтвержденные события PROBLEM, *Подтвержденные события PROBLEM*, *Неподтвержденные события RESOLVED*, *Подтвержденные события RESOLVED*	Введите новый код цвета или щелкните по цвету, чтобы выбрать новый из доступной палитры. Если установлен флажок *мигание*, триггеры будут некоторое время мигать при изменении состояния, чтобы стать более заметными.
Показывать OK-триггеры в течение	Период времени для отображения OK-триггеров. Допустимый диапазон: 0 - 24 часа. Поддерживаются суффиксы времени, например 5m, 2h, 1d.
При изменении состояния триггеры мигают в течение	Длительность мигания триггера. Допустимый диапазон: 0 - 24 часа. Поддерживаются суффиксы времени, например 5m, 2h, 1d.
Не классифицировано, *Информация*, *Предупреждение*, *Средняя*, *Высокая*, *Чрезвычайная*	Пользовательские имена и/или цвета серьезности для отображения вместо системных значений по умолчанию. Введите новый код цвета или щелкните по цвету, чтобы выбрать новый из доступной палитры. Обратите внимание, что введенные здесь пользовательские имена серьезности будут использоваться во всех локалях. Если вам нужно перевести их на другие языки для определенных пользователей, см. страницу Настройка серьезностей триггеров.

Географические карты

Этот раздел позволяет выбрать поставщика сервиса тайлов географической карты и настроить параметры поставщика для виджета панели мониторинга Geomap. Для визуализации с использованием географических карт Zabbix использует библиотеку интерактивных карт Leaflet с открытым исходным кодом на JavaScript. Обратите внимание, что Zabbix не контролирует качество изображений, предоставляемых сторонними поставщиками тайлов, включая предопределенных поставщиков тайлов.



Parameter	Description
Tile provider	Выберите одного из доступных поставщиков сервиса тайлов или выберите *Other*, чтобы добавить другого поставщика тайлов или самостоятельно размещенные тайлы (см. Использование пользовательского поставщика сервиса тайлов).
Tile URL	Шаблон URL (до 2048 символов) для загрузки и отображения слоя тайлов на географических картах. Это поле доступно для редактирования только если для *Tile provider* выбрано *Other*. Поддерживаются следующие заполнители: *{s}* представляет один из доступных поддоменов; *{z}* представляет параметр уровня масштабирования в URL; *{x}* и *{y}* представляют координаты тайла; *{r}* можно использовать для добавления "@2x" к URL, чтобы загружать тайлы для экранов Retina. Пример: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Текст атрибуции поставщика тайлов, который будет отображаться в небольшом текстовом поле на карте. Это поле видно только если для *Tile provider* выбрано *Other*.

Parameter	Description
Max zoom level	Максимальный уровень масштабирования карты. Это поле доступно для редактирования только если для *Tile provider* выбрано *Other*.

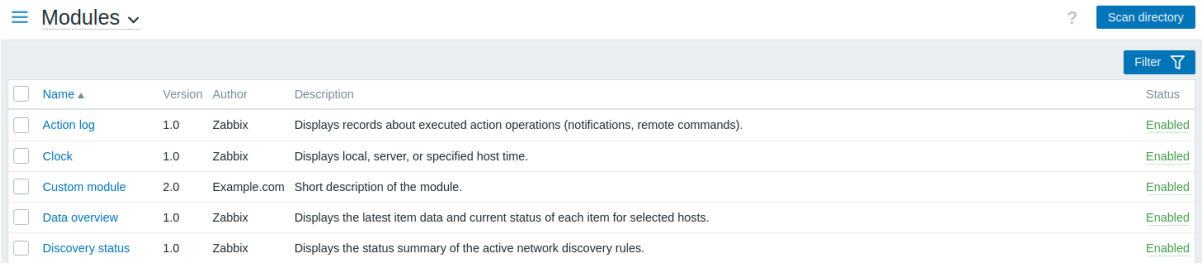
Использование пользовательского поставщика услуг плиток

Виджет *Геокарта (Geomap)* способен загружать растровые изображения плиток с пользовательского собственного хостинга либо службы стороннего поставщика плиток. Чтобы использовать службу стороннего поставщика плиток или собственную папку или сервер плиток, выберите *Другое (Other)* в поле *Провайдер плитки (Tile provider)* и укажите собственный URL в поле *URL плитки (Tile URL)*, используя соответствующие шаблоны-заполнители.

Модули

В этом разделе можно управлять пользовательскими и встроенными модулями веб-интерфейса.

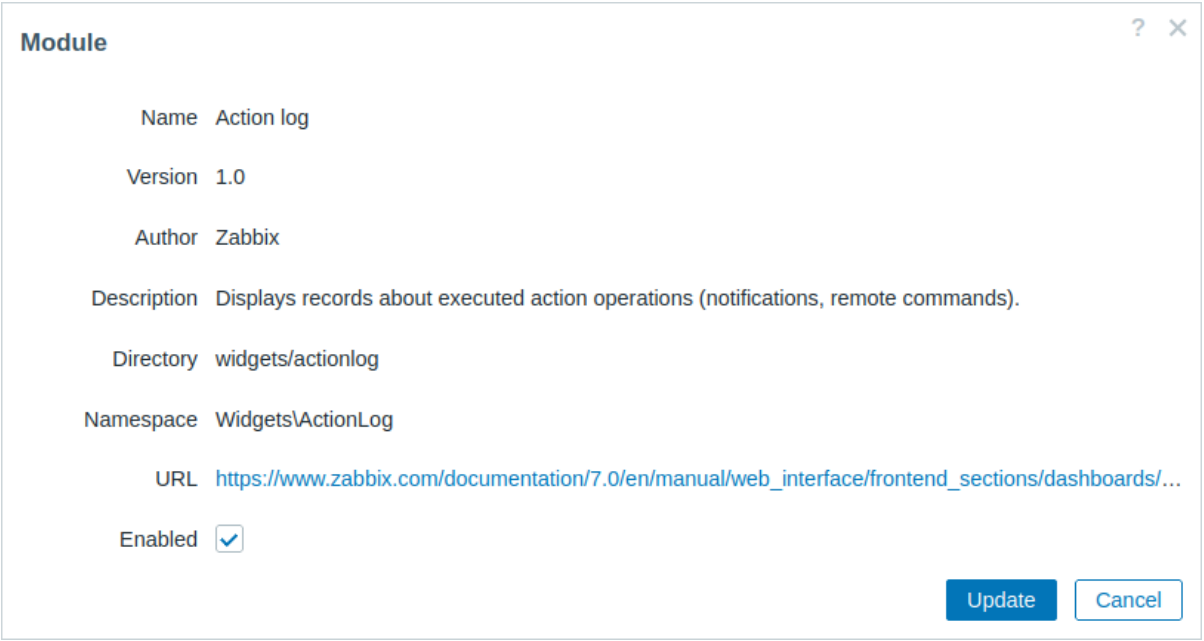
Note:
Если флаг функции `$ZBX_FEATURE_FLAGS['modules_config_enabled']` установлен в значение `false` в файле конфигурации веб-интерфейса, доступ к этому разделу отключен.



Нажмите *Сканировать каталог*, чтобы зарегистрировать или отменить регистрацию пользовательских модулей. Зарегистрированные модули появятся в списке, а незарегистрированные будут удалены из списка.

Нажмите на статус модуля в списке, чтобы включить или отключить модуль. Также можно массово включить или отключить модули: выберите их в списке, а затем нажмите кнопки *Включить* или *Отключить* под списком.

Нажмите на имя модуля в списке, чтобы просмотреть его подробные сведения во всплывающем окне.

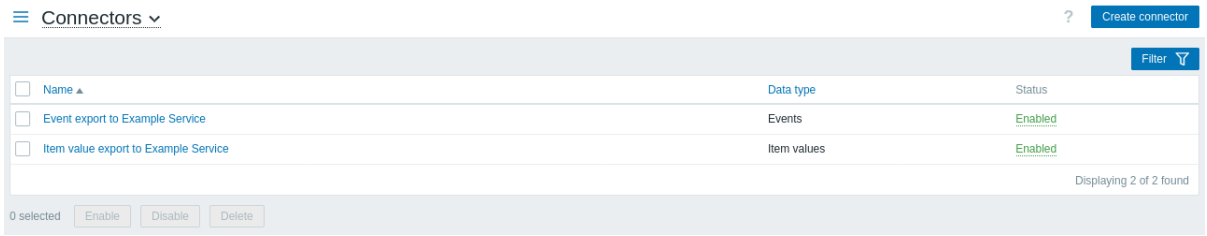


Статус модуля также можно изменить во всплывающем окне с подробными сведениями о модуле: установите или снимите флажок *Включен*, а затем нажмите *Обновить*.

Можно фильтровать модули по имени или статусу (включен/отключен).

Коннекторы

В этом разделе можно настроить коннекторы для потоковой передачи данных Zabbix во внешние системы по протоколу HTTP.



Нажмите *Создать коннектор*, чтобы настроить новый коннектор.

Коннекторы можно фильтровать по имени или статусу (включен/отключен). Нажмите на статус коннектора в списке, чтобы включить или отключить его. Также можно массово включить или отключить коннекторы: выберите их в списке, а затем нажмите кнопки *Включить/Отключить* под списком.

Другое

Этот раздел позволяет настроить прочие параметры веб-интерфейса.



Parameter	Description
Frontend URL	URL (до 2048 символов) для веб-интерфейса Zabbix. Этот параметр используется веб-службой Zabbix для связи с веб-интерфейсом и должен быть указан, чтобы включить запланированные отчеты.
Group for discovered hosts	Узлы сети, обнаруженные с помощью network discovery и agent autoregistration, будут автоматически помещены в группу узлов сети, выбранную здесь.

Parameter	Description
Default host inventory mode	Режим по умолчанию для инвентаризации узла сети. Он будет использоваться каждый раз при создании нового узла сети или прототипа узла сети сервером или веб-интерфейсом, если только он не будет переопределен во время обнаружения узла сети/авторегистрации с помощью операции *Set host inventory mode*.
User group for database down message	Группа пользователей для отправки сообщения тревоги о недоступности базы данных. Сервер Zabbix зависит от доступности внутренней базы данных. Он не может работать без базы данных. Если база данных недоступна, выбранные пользователи могут быть уведомлены Zabbix. Уведомления будут отправляться в указанную здесь группу пользователей с использованием включенных элементов пользовательских медиа. Уведомления передаются с использованием следующих типов медиа (если включены): email; SMS; пользовательские скрипты оповещения. Даже если элемент медиа типа вебхук настроен и включен, он не будет вызывать уведомления. Сервер Zabbix не остановится; он будет ждать восстановления базы данных, чтобы продолжить обработку. Уведомление состоит из следующего содержимого: `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> не добавляется в сообщение, если для него задано пустое значение, а <DB Port> не добавляется, если используется значение по умолчанию (”0”). Менеджер оповещений (специальный процесс сервера Zabbix) пытается установить новое соединение с базой данных каждые 10 секунд. Если база данных по-прежнему недоступна, менеджер оповещений повторяет отправку сообщений, но не чаще чем раз в 15 минут.
Log unmatched SNMP traps	Записывать SNMP trap в журнал, если не найдено соответствующих интерфейсов SNMP.

Авторизация

Параметр	Описание
Попытки входа (Login attempts)	Количество неуспешных попыток входа, прежде чем возможность входа будет заблокирована.
Интервал блокировки входа в систему (Login blocking interval)	Период времени, в течение которого вход в систему будет запрещён при превышении лимита *попыток входа*. Допустимый диапазон: 30 - 3600 секунд. Поддерживаются суффиксы времени, например: 90s, 5m, 1h.

Хранение секретов

Parameter	Description
Vault provider	Выберите программное обеспечение для управления секретами для хранения значений user macro - *HashiCorp Vault* (по умолчанию) или *CyberArk Vault*.
Resolve secret vault macros by	Разрешать макросы secret vault с помощью: **Zabbix server** - секреты извлекаются сервером Zabbix и при необходимости передаются прокси (по умолчанию); **Zabbix server and proxy** - секреты извлекаются как сервером Zabbix, так и прокси, что позволяет им независимо разрешать макросы.

См. также: Хранение секретов.

Безопасность

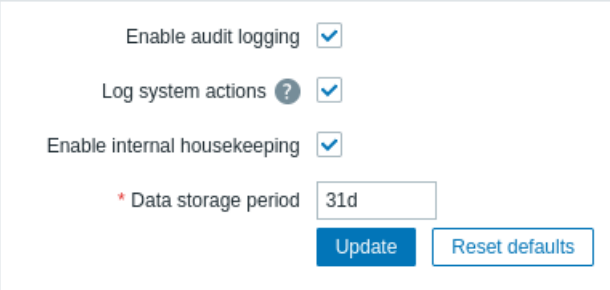
Параметр	Описание
Валидация схем URI (Validate URI schemes)	Снимите флажок, чтобы отключить проверку схемы URI (по умолчанию включено). Если включено, можно указать список (через запятую) разрешённых схем URI (по умолчанию: http,https,ftp,file,mailto,tel,ssh). Применяется ко всем полям в веб-интерфейсе, где используются URI (например, URL’ы элементов карт сети).

Параметр	Описание
HTTP заголовок X-Frame-Options (Use X-Frame-Options HTTP header)	Снимите данный флажок, чтобы отключить заголовок X-Frame-options (не рекомендуется). Если отмечено, вы можете указать значение HTTP заголовка X-Frame-options. Поддерживаемые значения: **SAMEORIGIN** (по умолчанию) или **'self'** (должно быть обрамлено апострофами) — страница может отображаться только внутри фрейма с тем же источником, как и сама страница; **DENY** или **'none'** (должно быть обрамлено апострофами) — страница не может отображаться во фрейме, независимо от сайта, пытающегося это сделать; **строка имён хостов через пробел**; добавление **'self'** (должно быть обрамлено апострофами) к списку позволяет странице отображаться во фрейме с тем же источником, как и сама страница. Обратите внимание, что использование **'self'** или **'none'** без апострофов приведёт к тому, что они будут рассматриваться как имена хостов.
Использование атрибута sandbox в iframe (Use iframe sandboxing)	Снимите данный флажок, чтобы не помещать полученное по URL содержимое в песочницу (не рекомендуется). Если отмечено, вы можете указать исключения песочницы iframe; неуказанные ограничения по-прежнему будут применяться. Если это поле пустое, применяются все ограничения атрибута sandbox. Смотрите описание атрибута `sandbox` [en]для более подробной информации.

Журнал аудита

Обзор

В этом разделе можно настроить параметры журнала аудита.



Доступны следующие параметры:

Параметр	Описание
Активировать журналирование аудита (Enable audit logging)	Включить (по умолчанию) или отключить ведение журнала аудита.
Журналирование действий системы (Log system actions)	Включить (по умолчанию) или отключить ведение журнала аудита низкоуровневого обнаружения, сетевого обнаружения и действий авторегистрации, выполняемых сервером (пользователем System).
Активировать внутреннюю очистку истории (Enable internal housekeeping)	Включить (по умолчанию) или отключить внутреннюю очистку истории для записей журнала аудита.
Период хранения данных (Data storage period)	Количество дней, в течение которых аудитные записи должны храниться, прежде чем они будут удалены процессом очистки истории. Обязательно, если функция очистки включена. По умолчанию: 31 день.

Параметры сжатия аудитных записей доступны в разделе *Администрирование → Очистка истории (Administration → Housekeeping)*, в блоке *Сжатие истории, тенденций и журнала аудита (History, trends and audit log compression)*, который становится видимым при использовании TimescaleDB.

Обслуживание

Обзор

Раздел *Administration* > *Housekeeping* позволяет настроить внутреннюю очистку Zabbix, которая автоматически удаляет устаревшие данные или данные, удаленные пользователем, из базы данных. Это предотвращает бесконечный рост базы данных, который приводил бы к ненужному расходу дискового пространства и снижению производительности базы данных.

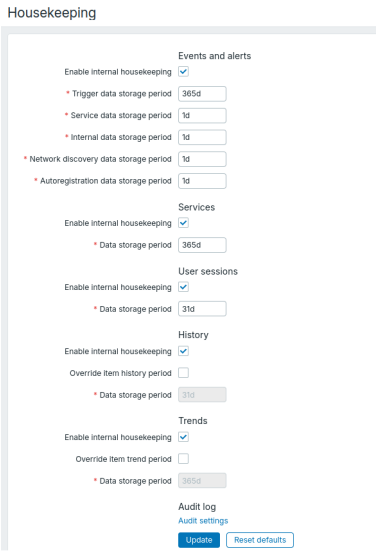
Очистку можно включить и настроить для следующих типов данных:

- События и оповещения от триггеров, сервисов, внутренних данных, сетевого обнаружения и авто-регистрации
- Сервисы
- Сеансы пользователей
- История и тренды для элементов данных

Note:
Очистка журнала аудита настраивается отдельно.

Конфигурация

Доступны следующие параметры:



Parameter	Description
Enable internal housekeeping	Включить или отключить внутреннюю очистку (по умолчанию включена). Если включено, процедура очистки автоматически удаляет из базы данных данные, срок хранения которых превышает *Data storage period*.
Data storage period	Укажите, как долго хранить данные до их удаления очистителем. Обязательно, если включена внутренняя очистка. Диапазон: 1 day (1 hour for history) - 25 years; или ”0”. Поддерживаются суффиксы времени (например, 1d, 1w). Для *Events and alerts* срок хранения данных задается отдельно для триггеров, сервисов, внутренних данных, обнаружения сети и авторегистрации. Очиститель удаляет только те события, которые не связаны с проблемами. Например, событие проблемы/восстановления старше *Data storage period* не будет удалено, если оно связано с нерешенной проблемой. Когда очиститель удаляет устаревшие сущности, он сначала удаляет проблемы, затем события. Для *History* и *Trends* сроки хранения данных также определяют, как долго данные остаются видимыми в *Monitoring* > *Latest data*, даже если внутренняя очистка отключена. Обратите внимание: когда элемент данных или триггер удаляется, его проблемы и связанные события также удаляются.

Parameter	Description
Override item history period	Если включено, срок хранения history/trend, указанный в конфигурации элемента данных, будет переопределен настройкой *Data storage period* (за исключением элементов данных с включенной опцией *Do not store*). Эту опцию можно использовать даже при отключенной внутренней очистке, если используется внешний очиститель.
Override item trend period	

Если используется **ClickHouse**, становится доступен раздел *ClickHouse data storage period*. В нем отображается срок хранения данных для каждого типа значений элемента данных. Чтобы изменить срок, см. Настройка срока хранения данных в ClickHouse. Обратите внимание, что очиститель не удаляет данные из ClickHouse.

Если используется **Elasticsearch**, становится доступен раздел *Elasticsearch data storage period*. В нем отображаются типы значений элементов данных, сохраненные в Elasticsearch, при этом срок хранения отображается как *Unknown*. Чтобы управлять временем хранения данных, см. Index Lifecycle Management. Обратите внимание, что очиститель не удаляет данные из Elasticsearch.

Если используется **TimescaleDB**, становится доступен раздел *History, trends and audit log compression*.

Note:
Для TimescaleDB включите *Override item history period*, *Override item trend period* и *Enable internal housekeeping* для history и trends, чтобы в полной мере использовать автоматическое разбиение на разделы. Если эти параметры отключены, данные, хранящиеся в таблицах history и trends, по-прежнему будут разбиваться на разделы, но очиститель не будет удалять устаревшие разделы, и будут появляться предупреждения о конфигурации. Когда удаление устаревших разделов включено, сервер Zabbix и веб-интерфейс больше не будут отслеживать удаленные элементы данных, а history для этих элементов будет очищена при удалении устаревшего раздела.

По умолчанию очиститель выполняется автоматически каждый час. Вы можете изменить частоту выполнения, задав конфигурационный параметр сервера Zabbix `HousekeepingFrequency`.

Кнопка *Reset defaults* позволяет отменить все внесенные изменения.

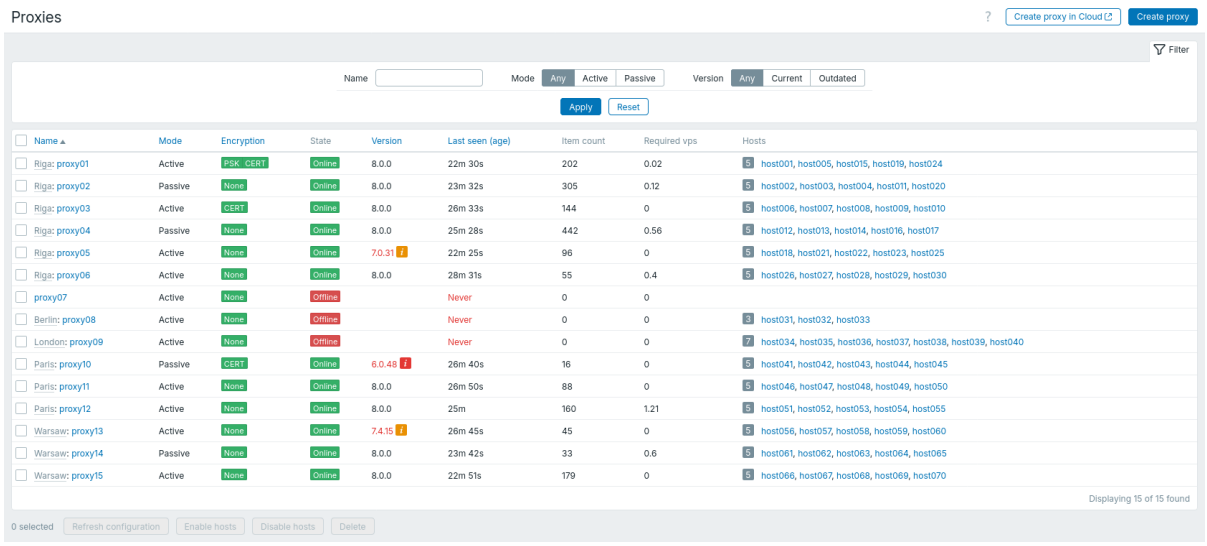
Прокси

Обзор

В разделе *Администрирование* > *Прокси* веб-интерфейса Zabbix можно настроить прокси для распределенного мониторинга.

Прокси

Отображается список существующих прокси с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя прокси. При нажатии на имя прокси открывается форма настройки прокси. Если прокси входит в группу прокси, перед именем прокси отображается имя группы в виде серой ссылки. При нажатии на имя группы открывается форма настройки группы прокси.
Режим	Режим прокси - *Активный* или *Пассивный*.
Шифрование	Состояние шифрования соединений от прокси: **Нет** - без шифрования; **PSK** - с использованием предварительно установленного ключа; **Сертификат** - с использованием сертификата.
Состояние	Состояние прокси: **Неизвестно** - прокси был создан, когда сервер Zabbix был недоступен, или сервер еще не обновил состояние; **В сети** - прокси связывался с сервером Zabbix в течение периода переключения при отказе (пассивный прокси ответил на запрос сервера; активный прокси отправил запрос); **Не в сети** - прокси не связывался с сервером Zabbix в течение периода переключения при отказе.
Версия	Версия прокси (трехзначный номер версии). Если прокси устарел или не поддерживается, номер версии выделяется (красным цветом), а значок состояния информации (желтый или красный) отображается. Наведите указатель мыши на значок, чтобы просмотреть подробную информацию.
Последний контакт (возраст)	Время, когда сервер в последний раз обнаружил прокси.
Количество элементов данных	Количество включенных элементов данных на включенных узлах сети, назначенных прокси.
Требуемая производительность	Требуемая производительность прокси (количество значений, которые необходимо собирать в секунду).
Узлы сети	Количество включенных узлов сети, назначенных прокси, и список узлов сети, мониторинг которых выполняется прокси. При нажатии на имя узла сети открывается форма настройки узла сети.

Чтобы настроить новый прокси, нажмите кнопку *Создать прокси* в правом верхнем углу. Чтобы развернуть новый прокси в Zabbix Cloud, нажмите расположенную рядом кнопку *Создать прокси в Cloud*. В новой вкладке браузера откроется Zabbix Cloud, где можно создать узел прокси. После создания узла прокси настройте прокси с помощью кнопки *Создать прокси*. Дополнительную информацию см. в разделе Настройка прокси Zabbix Cloud.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

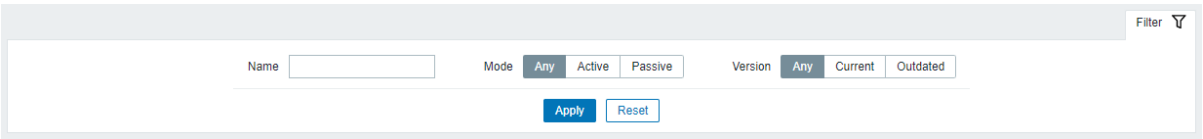
- *Обновить конфигурацию* - обновить конфигурацию прокси;
- *Включить узлы сети* - изменить статус узлов сети, контролируемых прокси, на *Контролируется*;
- *Отключить узлы сети* - изменить статус узлов сети, контролируемых прокси, на *Не контролируется*;
- *Удалить* - удалить прокси.

Чтобы использовать эти параметры, отметьте флажки перед нужными прокси, затем нажмите требуемую кнопку.

Использование фильтра

С помощью фильтра можно отображать только интересующие вас прокси. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком прокси. При нажатии на неё становится доступен фильтр, с помощью которого можно фильтровать прокси по имени, режиму и версии. Обратите внимание, что параметр фильтра *Устаревшие* отображает как устаревшие (частично поддерживаемые), так и неподдерживаемые прокси.



Группы прокси

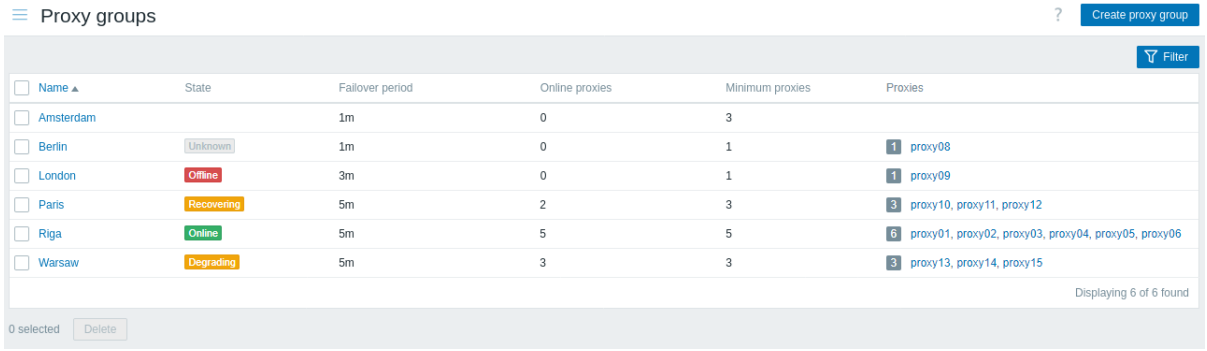
Обзор

В *Administration* > *Proxy groups* можно настроить группы прокси.

Группы прокси используются в балансировке нагрузки прокси с автоматическим распределением узлов сети между прокси и обеспечением высокой доступности между прокси.

Группы прокси

Отображается список существующих групп прокси с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы прокси. При нажатии на имя группы прокси открывается форма настройки группы прокси.
Состояние	Состояние группы прокси (отображается, если группа содержит хотя бы один прокси): **Неизвестно** — группа была создана, когда сервер Zabbix был недоступен, или сервер еще не обновил состояние; **В сети** — минимальное количество прокси связалось с сервером Zabbix в течение периода переключения при отказе; **Деградация** — группа вскоре перейдет в автономный режим, поскольку количество прокси в сети стало ниже порогового значения *Минимальное количество прокси*; **Не в сети** — с сервером Zabbix в течение периода переключения при отказе связалось меньше минимального количества прокси; **Восстановление** — группа вскоре вернется в сеть, поскольку количество прокси в сети достигло порогового значения *Минимальное количество прокси*. Состояние группы прокси можно отслеживать с помощью элементов данных zabbix[proxy group,<name>,state] и zabbix[proxy group,discovery].
Период переключения при отказе	Период, в течение которого прокси в группе прокси должен связаться с сервером Zabbix, чтобы считаться находящимся в сети.
Прокси в сети	Количество прокси в сети (отображается красным цветом, если оно ниже минимального количества для группы).
Минимальное количество прокси	Минимальное количество прокси в сети, необходимое для поддержания группы прокси в сети.
Прокси	Количество прокси в группе и список прокси в группе со ссылками на форму настройки прокси. Максимальное количество отображаемых прокси ограничено значением *Максимальное количество элементов для отображения внутри ячейки таблицы*.

Чтобы настроить новую группу прокси, нажмите кнопку *Создать группы прокси* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

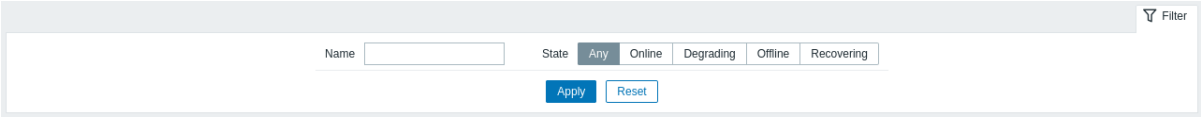
- *Удалить* - удалить группы прокси.

Чтобы использовать эти параметры, отметьте флажки перед соответствующими группами прокси, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те группы прокси, которые вам интересны. Для повышения производительности поиска данные ищутся с неразрешенными макросами.

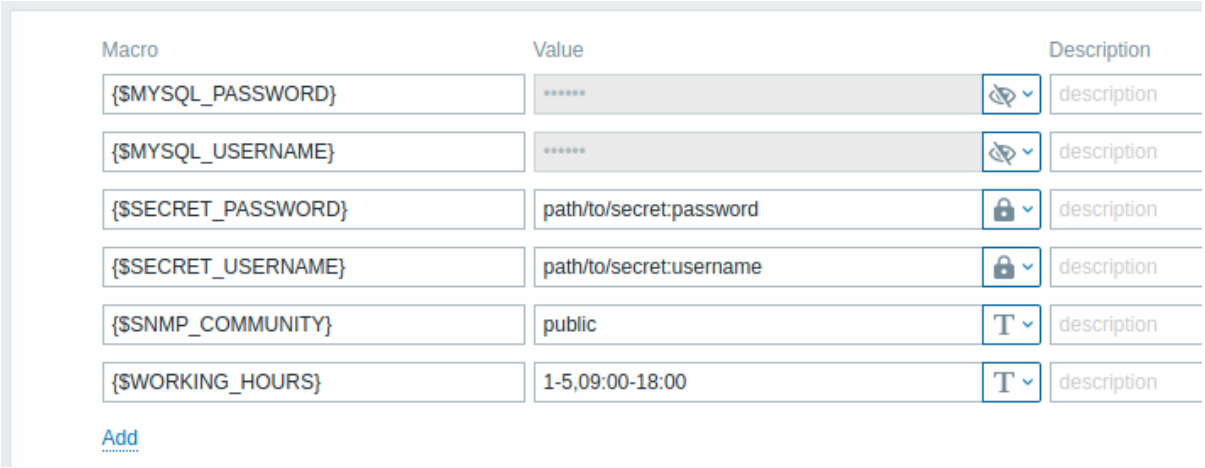
Ссылка *Filter* доступна над списком групп прокси. Если вы нажмете ее, станет доступен фильтр, с помощью которого можно фильтровать группы прокси по имени и статусу.



Макросы

Обзор

В этом разделе можно определить пользовательские макросы на уровне всей системы в виде пар «имя — значение». Обратите внимание, что значения макросов можно хранить как обычный текст, секретный текст или секрет в Vault. Также можно добавить описание.



Очередь

Обзор

В разделе *Администрирование* > *Очередь* отображаются элементы данных, ожидающие обновления.

В идеальном случае при открытии этого раздела все должно быть «зеленым», то есть в очереди не должно быть элементов данных. Если все элементы данных обновляются без задержек, ожидающих элементов нет. Однако из-за недостаточной производительности сервера обновление некоторых элементов данных может задерживаться, и соответствующая информация отображается в этом разделе. Дополнительные сведения см. в разделе Очередь.

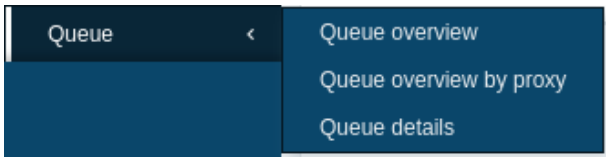
Note:
Прокси Zabbix не учитывает периоды обслуживания; подробные сведения см. в разделе Расчет очередей во время обслуживания.

Note:
Очередь доступна только при работающем сервере Zabbix. Элементы данных не учитываются в очереди, если интерфейс элемента данных становится недоступным из-за проблем с подключением или неправильной работы агента.

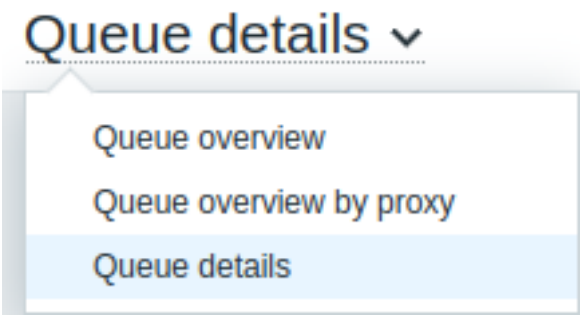
Раздел *Администрирование* > *Очередь* содержит следующие страницы:

- Обзор очереди - отображает очередь по типу элемента данных;
- Обзор очереди по прокси - отображает очередь по прокси;
- Подробности очереди - отображает список элементов данных с задержкой.

Список доступных страниц появляется при нажатии *Очередь* в разделе меню *Администрирование*. Также можно переключаться между страницами с помощью раскрывающегося списка заголовков в левом верхнем углу.



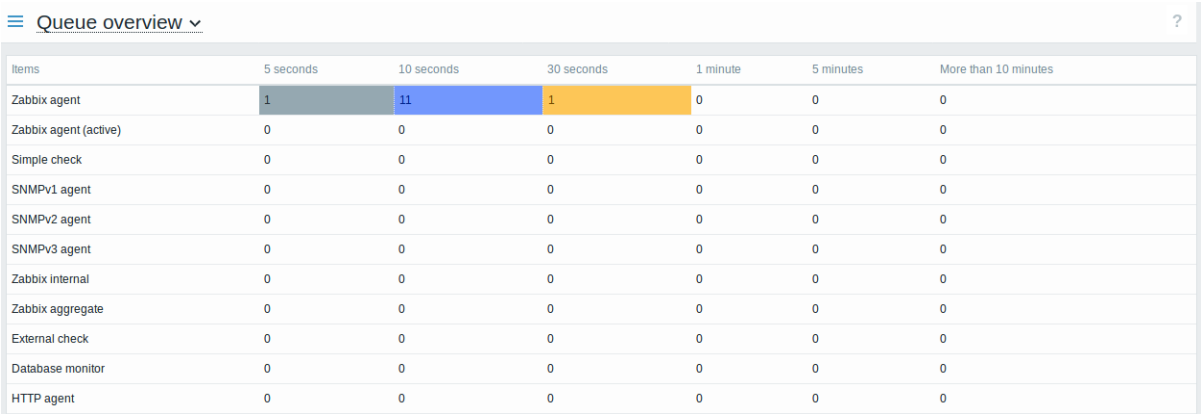
Меню третьего уровня.



Раскрывающийся список заголовков.

Обзор по типу элементов данных

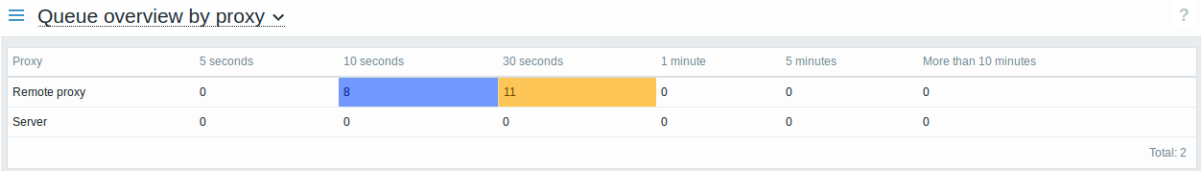
На этом экране можно легко определить, связана ли проблема с одним или несколькими типами элементов данных.



Каждая строка содержит тип элемента данных. Каждый столбец отображает количество ожидающих элементов данных: ожидающих 5—10 секунд / 10—30 секунд / 30—60 секунд / 1—5 минут / 5—10 минут и более 10 минут соответственно.

Обзор по прокси

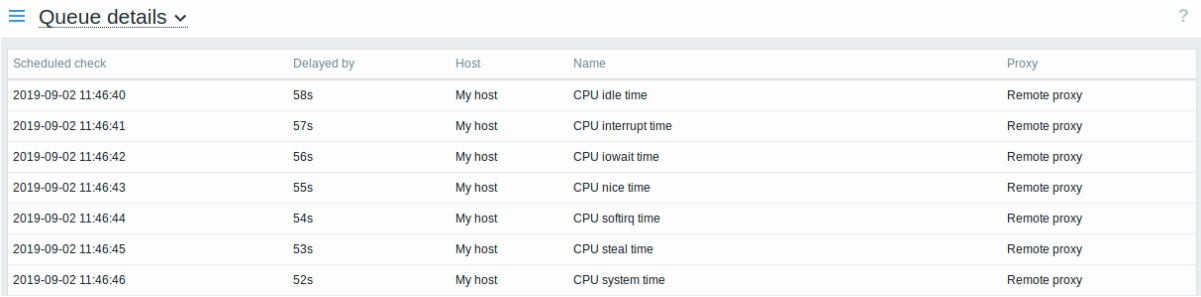
На этом экране можно легко определить, связана ли проблема с одним из прокси или с сервером.



Каждая строка содержит один прокси, и последняя строка в списке — сервер. Каждый столбец отображает количество ожидающих элементов данных: ожидающих 5—10 секунд / 10—30 секунд / 30—60 секунд / 1—5 минут / 5—10 минут и более 10 минут соответственно.

Список ожидающих элементов данных

На этом экране перечислены все ожидающие элементы данных.



Отображаемые данные:

Столбец	Описание
Запланированная проверка (Scheduled check)	Время, когда проверка должна была выполниться.
Задерживается на (Delayed by)	Длительность задержки.
Узел сети (Host)	Узел сети, которому принадлежит элемент данных.
Имя (Name)	Имя ожидающего элемента данных.
Прокси (Proxy)	Имя прокси-сервера, если узел сети наблюдается через прокси.

Возможные сообщения об ошибках

Возможно возникновение ситуации, когда данные не отображаются и появляется следующее сообщение об ошибке:



В этом случае отображается следующее сообщение об ошибке:

```
Cannot display item queue. Permission denied
```

Это происходит, когда параметры конфигурации PHP в файле конфигурации веб-интерфейса - `$ZBX_SERVER` или оба параметра `$ZBX_SERVER` и `$ZBX_SERVER_PORT` - указывают на существующий сервер Zabbix, использующий другую базу данных.

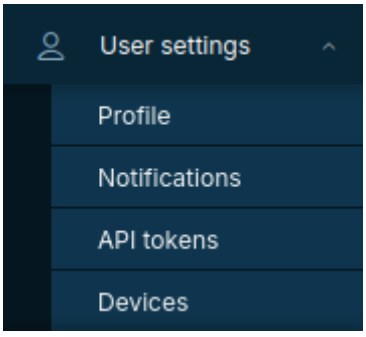
Настройки пользователя

Обзор

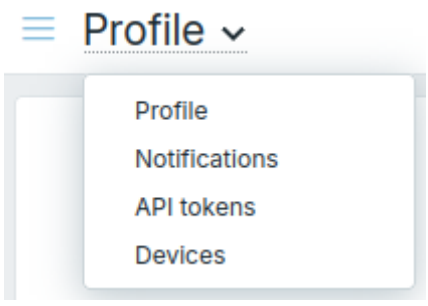
В зависимости от вашей роли пользователя, раздел *Настройки пользователя* может содержать следующие страницы:

- *Профиль* или *Профиль пользователя* - для настройки определенных функций веб-интерфейса Zabbix;
- *Уведомления* - для настройки уведомлений текущего пользователя;
- *Токены API* - для управления токенами API, назначенными текущему пользователю;
- *Устройства* - для управления мобильными устройствами, принадлежащими текущему пользователю.

Список доступных страниц отображается при нажатии значка пользователя в нижней части меню Zabbix (недоступно для пользователя *guest*). Также можно переключаться между страницами с помощью раскрывающегося списка заголовка в левом верхнем углу.



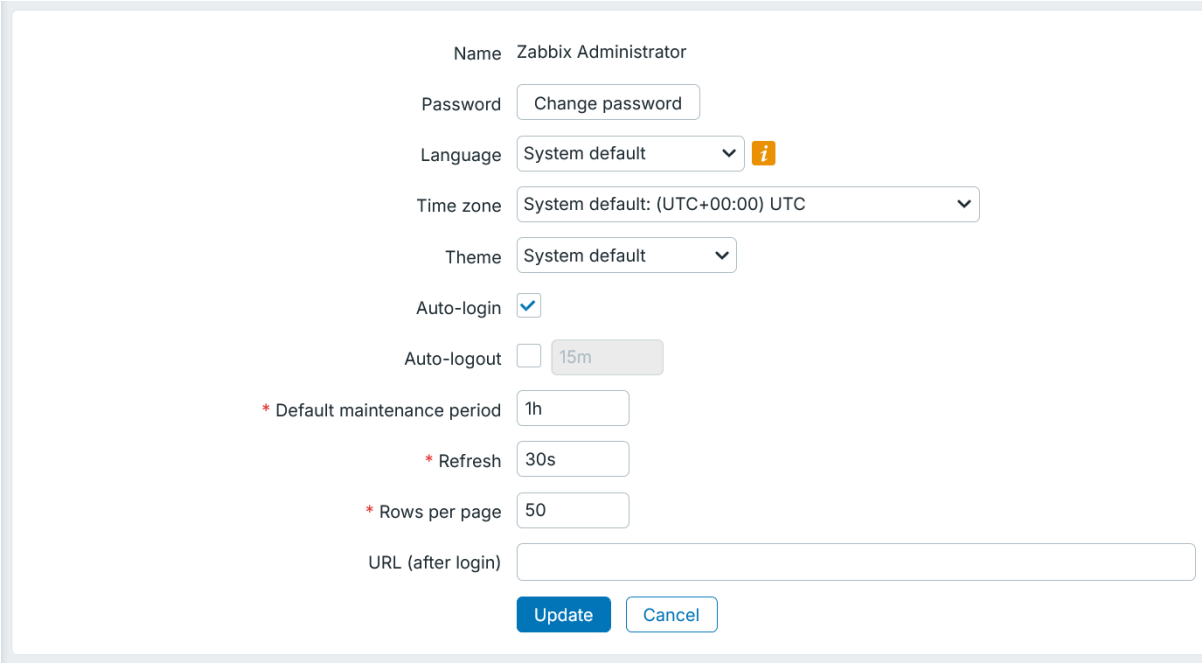
Меню третьего уровня.



Раскрывающийся список заголовка.

Профиль пользователя

Раздел *Профиль пользователя* содержит параметры для выбора языка веб-интерфейса, цветовой темы, количества строк, отображаемых в списках, и т. д. Изменения, внесенные здесь, применяются только к текущему пользователю.



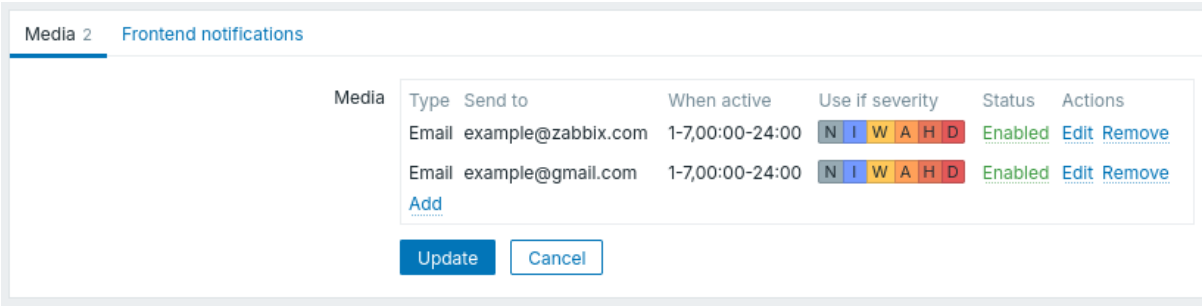
Параметр	Описание
Имя	Отображаются имя и фамилия пользователя. Если имя и фамилия не указаны, отображается имя пользователя.
Пароль	Нажмите кнопку *Изменить пароль*, чтобы открыть три поля: *Старый пароль*, *Новый пароль*, *Новый пароль (еще раз)*. После успешного изменения пароля пользователь будет отключен от всех активных сеансов. Обратите внимание, что пароль можно изменить только для пользователей, использующих внутреннюю аутентификацию Zabbix.
Язык	Выберите нужный язык веб-интерфейса или выберите **Системный по умолчанию**, чтобы использовать системные настройки по умолчанию. При выборе *Английский (en_US)* в веб-интерфейсе также будет включен американский формат времени и даты. Дополнительную информацию см. в разделе Установка дополнительных языков веб-интерфейса.
Часовой пояс	Выберите часовой пояс, чтобы переопределить глобальный часовой пояс на уровне пользователя, или выберите **Системный по умолчанию**, чтобы использовать глобальные настройки часового пояса.
Тема	Выберите цветовую тему для своего профиля: **Системная по умолчанию** — использовать системные настройки по умолчанию; **Синяя** — стандартная синяя тема; **Синяя (классическая)** — стандартная синяя тема с использованием устаревших шрифтов; **Темная** — альтернативная темная тема; **Темно-синяя** — темно-синяя тема; **Темная (классическая)** — альтернативная темная тема с использованием устаревших шрифтов; **Светлая с высокой контрастностью** — светлая тема с высокой контрастностью; **Темная с высокой контрастностью** — темная тема с высокой контрастностью.
Автоматический вход	Установите этот флажок, чтобы Zabbix запомнил вас и автоматически выполнял вход в течение 30 дней. При входе с включенным параметром *Запомнить на 30 дней*: - Таймер автоматического выхода сбрасывается (сеанс сохраняется до окончания 30-дневного периода). - При последующих посещениях в течение 30 дней включается автоматический вход. При входе без параметра *Запомнить на 30 дней*: - Параметр автоматического входа очищается. - Автоматический выход выполняется в соответствии с настроенным временем ожидания. Для этого используются файлы cookie браузера.

Параметр	Описание
Автоматический выход	Если этот флажок установлен, автоматический выход будет выполнен по истечении заданного количества секунд (минимум 90 секунд, максимум 1 день). Поддерживаются суффиксы времени, например: 90s, 5m, 2h, 1d. Обратите внимание, что этот параметр не работает в следующих случаях: * Когда страницы меню *Мониторинг* выполняют фоновое обновление информации. Если открыты страницы, на которых данные обновляются через определенный интервал (панели мониторинга, графики, последние данные и т. д.), время жизни сеанса соответственно продлевается, отключая функцию автоматического выхода. * Если вход выполнен с установленным параметром *Запомнить меня на 30 дней*; в этом случае таймер автоматического выхода переопределяется, и сеанс остается активным в течение всего периода. * Если аутентификация выполняется через поставщика удостоверений SSO (IdP), автоматический выход завершает только сеанс Zabbix. Сеанс SSO с IdP остается активным, поэтому пользователь может снова войти без ввода пароля. Параметр *Автоматический выход* также может принимать значение «0», что означает отключение функции автоматического выхода после обновления настроек профиля.
Период обслуживания по умолчанию	Задайте период обслуживания по умолчанию для внепланового обслуживания.
Обновление	Задайте частоту обновления информации на страницах меню *Мониторинг* (минимум 0 секунд, максимум 1 час). Поддерживаются суффиксы времени, например: 30s, 90s, 1m, 1h.
Строк на странице	Задайте количество строк, отображаемых на странице в списках. Меньшее количество строк (и меньшее количество отображаемых записей) обеспечивает более быструю загрузку.
URL (после входа)	Задайте URL, который будет отображаться после входа. Вместо *Панелей мониторинга* можно указать, например, URL страницы *Мониторинг* > *Триггеры*. Это поле отключено, если пользователь *Суперадминистратор* ограничил URL после входа определенной страницей для вашей роли пользователя. Если поле отключено или пользователь не указал значение, под полем отображается значение по умолчанию, например `zabbix.php?action=dashboard.view`

Уведомления

Раздел *Уведомления* предоставляет параметры для настройки уведомлений текущего пользователя.

Вкладка *Способы оповещения* позволяет указать сведения о способах оповещения для пользователя, например типы способов оповещения и адреса, которые следует использовать, а также время их использования для доставки уведомлений.



Если тип способа оповещения отключен:

- после имени отображается желтый значок информации;
- в столбце *Состояние* отображается ”Отключено”.

Note:
Права пользователя на изменение собственных сведений о способах оповещения могут быть предоставлены/отозваны в зависимости от его роли пользователя (см. параметр *Создание и редактирование собственных способов оповещения*). Права пользователя Super admin на изменение сведений о способах оповещения других пользователей также могут быть предоставлены/отозваны в зависимости от его роли пользователя (см. параметр *Создание и редактирование способов оповещения пользователей*)

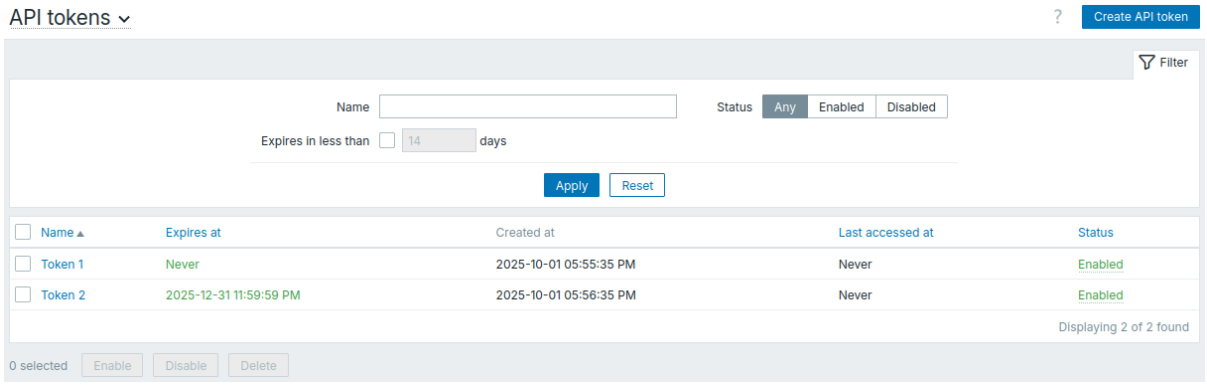
Обратите внимание, что для подготовленных пользователей:

- подготовленные способы оповещения пользователя нельзя удалить;
- подготовленные способы оповещения пользователя можно отключать/включать;
- такие поля подготовленных способов оповещения пользователя, как *Когда активно*, *Использовать при важности* и *Включено*, можно редактировать вручную;
- для подготовленных пользователей можно вручную добавлять дополнительные способы оповещения пользователя (например, дополнительные адреса электронной почты);
- способы оповещения пользователя, добавленные вручную, можно удалить.

Вкладка *Уведомления веб-интерфейса* позволяет настроить глобальные уведомления.

API tokens

Раздел *API tokens* позволяет просматривать токены, назначенные пользователю, редактировать сведения о токене и создавать новые токены. Этот раздел доступен пользователю только в том случае, если в настройках роли пользователя разрешено действие *Manage API tokens*.



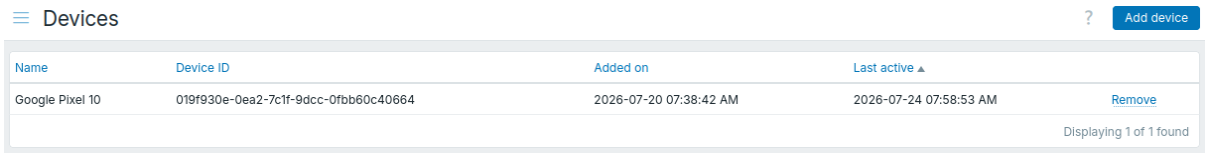
Вы можете фильтровать API tokens по имени, дате истечения срока действия или статусу (*Enabled/Disabled*). Щелкните статус токена в списке, чтобы быстро включить или отключить токен. Вы также можете включать или отключать несколько токенов одновременно, выбрав их в списке и затем нажав кнопки *Enable/Disable* под списком.

Attention:
Пользователи не могут просматривать значение *Auth token* токенов, назначенных им в Zabbix. Значение *Auth token* отображается только один раз — сразу после создания токена. Если токен был утерян, его необходимо сгенерировать заново.

Устройства

Раздел *Устройства* позволяет добавлять или удалять мобильные устройства для использования с мобильным приложением Zabbix.

Этот раздел отображается, если включен параметр `EnableMobileDevices`.



Чтобы добавить устройство:

1. Нажмите *Добавить устройство*.
2. Отсканируйте QR-код в мобильном приложении Zabbix.

Note:
При добавлении устройства Zabbix также добавляет тип носителя push-уведомления в носители пользователя. При удалении устройства этот носитель пользователя также удаляется. Если настроено несколько типов носителей push-уведомлений, Zabbix добавляет включенный тип, который первым идет в алфавитном порядке.

Добавленные устройства отображаются в списке. В списке указаны их имена и идентификаторы, а также даты и время добавления и последней активности.

Вы также можете удалять устройства. При удалении устройства его учетные данные для доступа отзываются и удаляются из Zabbix, после чего устройство больше не сможет получить доступ к Zabbix.

Суперадминистраторы также могут управлять устройствами других пользователей в разделе *Пользователи* > *Устройства*.

Глобальные уведомления

Обзор

Глобальные уведомления предоставляют способ отображать проблемы в реальном времени непосредственно на вашем текущем экране в веб-интерфейсе Zabbix.

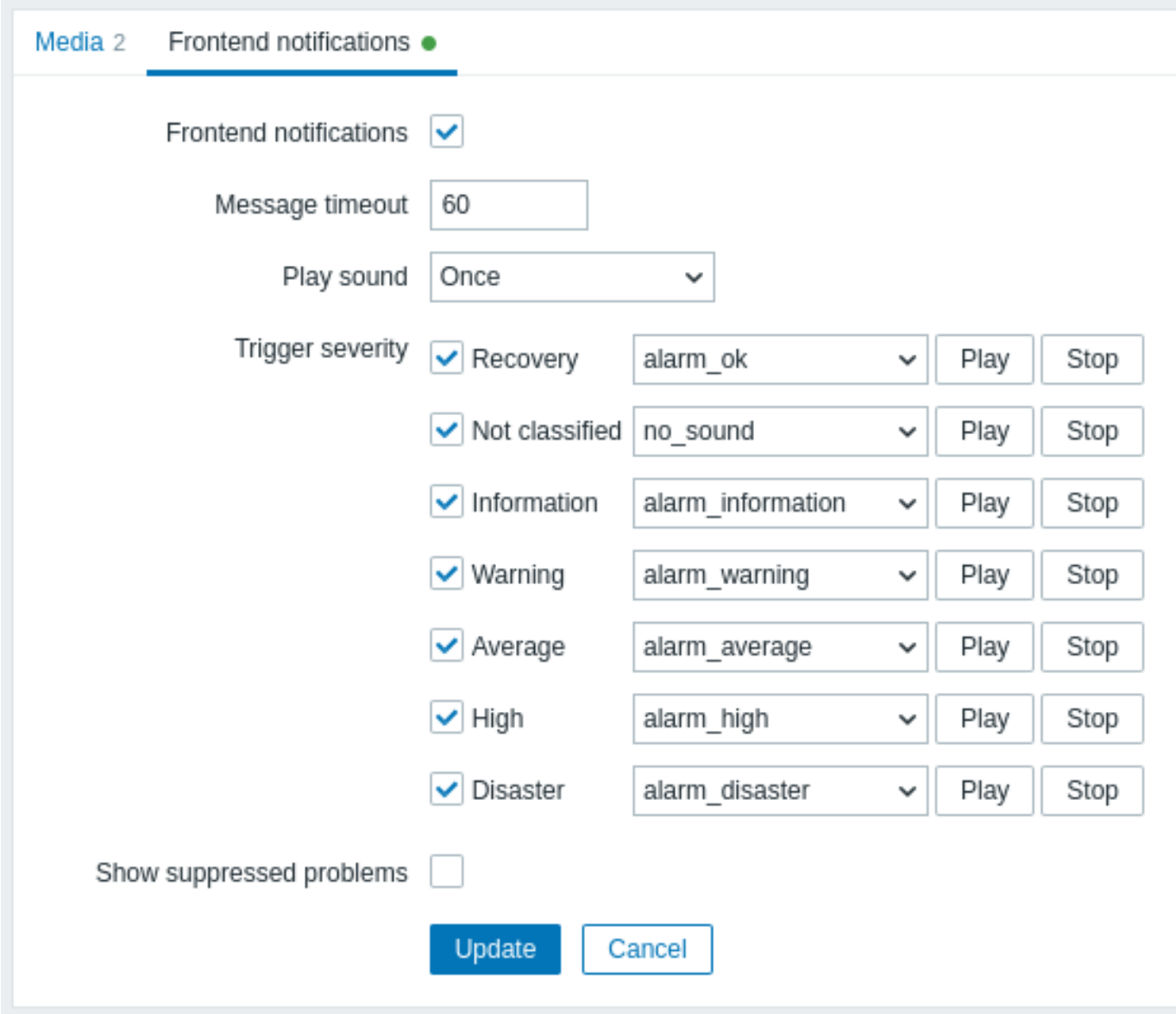
Без глобальных уведомлений при работе вне разделов *Проблемы* или *Дашборд* вы не получали бы никакой информации о текущих проблемах. Глобальные уведомления обеспечивают отображение этой информации независимо от того, где вы находитесь в веб-интерфейсе Zabbix.

Глобальные уведомления включают как отображение сообщения, так и воспроизведение звука.

Attention:
Автовоспроизведение звуков может быть отключено (по умолчанию) в последних версиях браузеров. В таких случаях необходимо включить этот параметр вручную.

Конфигурация

Глобальные уведомления можно включить для каждого пользователя на вкладке **Уведомления веб-интерфейса** в разделе Уведомления.

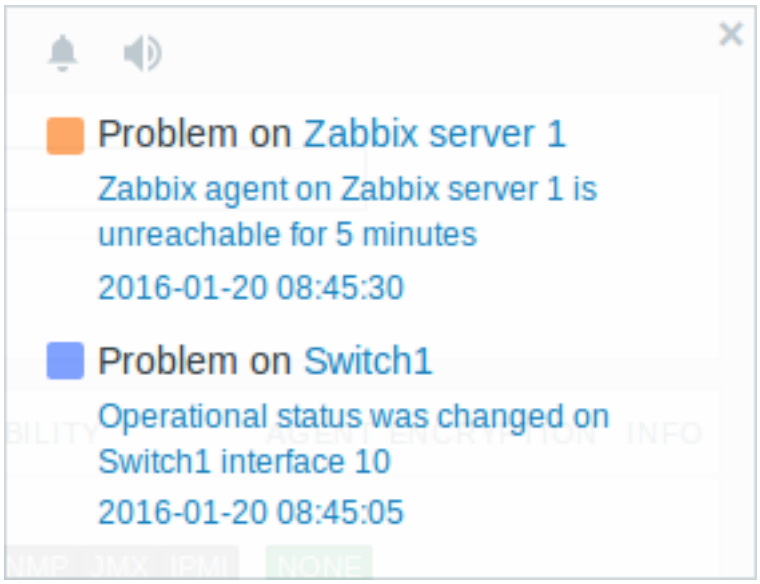


Parameter	Description
Frontend notifications	Установите флажок, чтобы включить глобальные уведомления.

Parameter	Description
Message timeout	Задайте длительность, в течение которой будет отображаться сообщение. По умолчанию сообщения остаются на экране 60 секунд. Поддерживаются суффиксы времени, например: 30s, 5m, 2h, 1d.
Play sound	Задайте длительность, в течение которой будет воспроизводиться звук. **Once** - звук воспроизводится один раз полностью; **10 seconds** - звук повторяется в течение 10 секунд; **Message timeout** - звук повторяется, пока сообщение отображается.
Trigger severity	Задайте уровни серьезности триггеров, для которых будут активироваться глобальные уведомления и звуки. Вы также можете выбрать звуки, подходящие для различных уровней серьезности. Если не отмечен ни один уровень серьезности, сообщения отображаться не будут. Кроме того, сообщения о восстановлении будут отображаться только для отмеченных уровней серьезности. Например, если отмечены *Recovery* и *Disaster*, глобальные уведомления будут отображаться для проблем и восстановлений триггеров с уровнем серьезности *Disaster*.
Show suppressed problems	Установите флажок, чтобы отображать уведомления о проблемах, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.

Глобальные сообщения, отображаемые

По мере поступления сообщений они отображаются в плавающем разделе справа. Вы можете свободно перемещать этот раздел, перетаскивая его заголовок.



Для этого раздела доступны следующие элементы управления:

- Кнопка **Snooze** отключает текущий активный звуковой сигнал тревоги;
- Кнопка **Mute/Unmute** переключает между воспроизведением и полным отключением звуковых сигналов тревоги.

Звук в браузерах

Обзор

Звук используется в глобальных уведомлениях.

Чтобы звуки воспроизводились во веб-интерфейсе Zabbix, в профиле пользователя на вкладке *Уведомления веб-интерфейса* должны быть включены *Уведомления веб-интерфейса*, при этом должны быть отмечены все уровни серьезности триггеров. Кроме того, звук должен быть включен во всплывающем окне глобальных уведомлений.

Если по какой-либо причине на устройстве невозможно воспроизведение звука, кнопка во всплывающем окне глобальных уведомлений будет постоянно находиться в состоянии ”mute”, а при наведении на кнопку будет

отображаться сообщение ”Cannot support notification audio for this device”.

Звуки, включая стандартные аудиоклипы, поддерживаются только в формате MP3.

Звуки веб-интерфейса Zabbix успешно протестированы в последних версиях браузеров Firefox и Opera в Linux, а также в браузерах Chrome, Firefox, Microsoft Edge и Opera в Windows.

Attention:
Автовоспроизведение звуков в последних версиях браузеров может быть отключено по умолчанию. В таких случаях эту настройку необходимо включить вручную.

Глобальный поиск

В веб-интерфейсе Zabbix можно выполнять поиск по узлам сети, группам узлов сети, шаблонам и группам шаблонов.

Поле ввода поиска находится под логотипом Zabbix в меню. Поиск можно запустить, нажав *Enter* или щелкнув значок поиска .



Если есть узел сети, в названии которого в любой части содержится введенная строка, появится выпадающий список со всеми такими узлами сети (совпадающая часть будет выделена оранжевым цветом). В выпадающем списке также будет показан узел сети, если его видимое имя совпадает с введенной строкой поиска по техническому имени; такой узел сети будет указан без какого-либо выделения.

Поисковые атрибуты

Узлы сети можно искать по следующим свойствам:

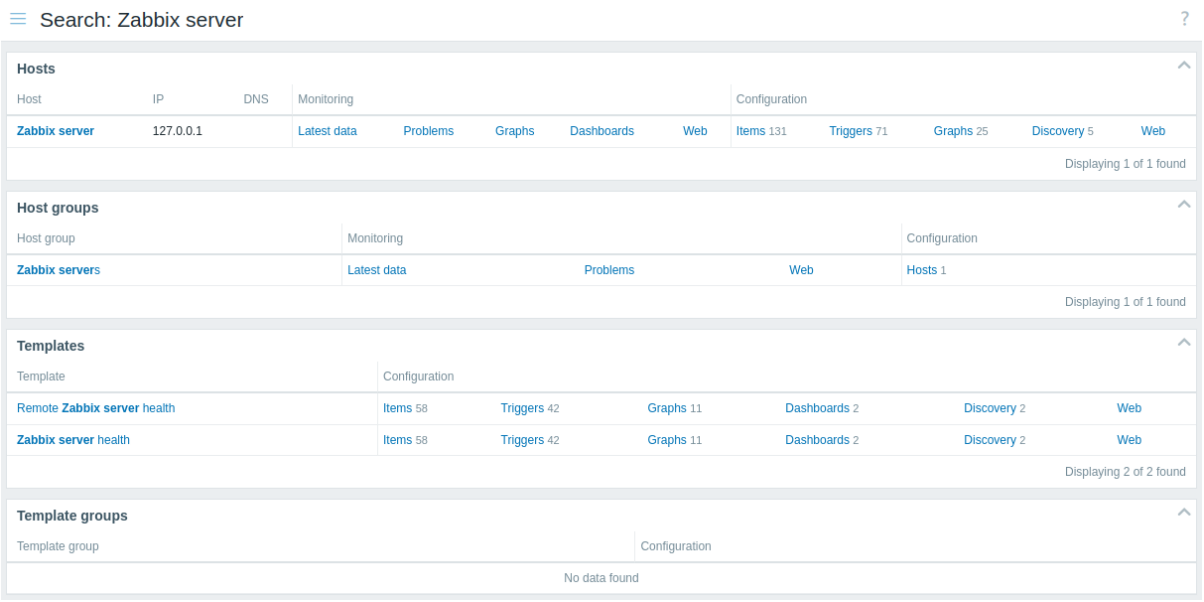
- Имя узла сети
- Видимое имя
- IP-адрес
- DNS-имя

Шаблоны можно искать по имени или видимому имени. Если вы выполняете поиск по имени, которое отличается от видимого имени (шаблона/узла сети), в результатах поиска оно отображается под видимым именем в скобках.

Группы узлов сети и шаблонов можно искать по имени. Указание родительской группы неявно выбирает все вложенные группы.

Результаты поиска

Результаты поиска состоят из четырех отдельных блоков для узлов сети, групп узлов сети, шаблонов и групп шаблонов.



Каждый отдельный блок можно свернуть/развернуть. Количество записей отображается в нижней части каждого блока, например, *Отображается 13 из 13 найденных*. Если записей нет, количество записей не отображается. Общее количество записей, отображаемых в одном блоке, ограничено 100.

Каждая запись содержит ссылки на данные мониторинга и конфигурации. См. полный список ссылок.

Для всех данных конфигурации (таких как элементы данных, триггеры, графики) количество найденных сущностей отображается числом рядом с названием сущности, серым цветом. **Примечание**: если сущностей нет, число не отображается.

Включенные узлы сети отображаются синим цветом, отключенные узлы сети — красным.

Доступные ссылки

Для каждой записи доступны следующие ссылки:

- Узлы сети
  - Мониторинг
    - Последние данные
    - Проблемы
    - Графики
    - Панели узла сети
    - Веб-сценарии
  - Конфигурация
    - Элементы данных
    - Триггеры
    - Графики
    - Правила обнаружения
    - Веб-сценарии
- Группы узлов сети
  - Мониторинг
    - Последние данные
    - Проблемы
    - Веб-сценарии
  - Конфигурация
    - Узлы сети
- Шаблоны
  - Конфигурация
    - Элементы данных
    - Триггеры
    - Графики
    - Панели шаблона
    - Правила обнаружения
    - Веб-сценарии
- Группы шаблонов
  - Конфигурация
    - Шаблоны

Режим обслуживания веб-интерфейса

Обзор

Можно временно отключить веб-интерфейс Zabbix, чтобы ограничить доступ. Это полезно для защиты базы данных Zabbix от изменений, инициированных пользователями, и сохранения ее целостности.

Пока веб-интерфейс Zabbix находится в режиме обслуживания, вы можете безопасно остановить базу данных и выполнить работы по обслуживанию.

Пользователи с указанных IP-адресов смогут нормально взаимодействовать с веб-интерфейсом во время режима обслуживания.

Настройка

Чтобы включить режим обслуживания, откройте файл `maintenance.inc.php` (расположенный в каталоге `/conf` каталога HTML-документов Zabbix на веб-сервере) и раскомментируйте следующие строки:

```
//            .
define('ZBX_DENY_GUI_ACCESS', 1);

//     IP-    ,                        -       (       ).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

//      ,                          (       ).
$ZBX_GUI_ACCESS_MESSAGE = '                MySQL   15:00.                ...';
```

Note:
В большинстве случаев файл `maintenance.inc.php` находится в каталоге `/conf` каталога HTML-документов Zabbix на веб-сервере. Однако в некоторых операционных системах и веб-серверах может использоваться другое расположение.
Например, файл находится по следующему пути:

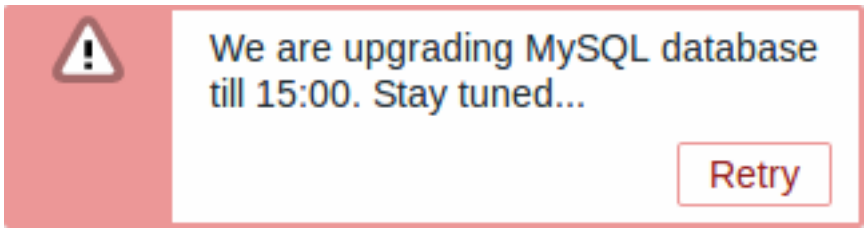
- В SUSE и RedHat: `/etc/zabbix/web/maintenance.inc.php`.
- В системах на базе Debian: `/usr/share/zabbix/conf/`.

См. также Копирование файлов PHP.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Если параметр определен с любым значением, будет включен режим обслуживания. Чтобы отключить режим обслуживания, закомментируйте или удалите параметр.
ZBX_GUI_ACCESS_IP_RANGE	Массив IP-адресов, которым разрешено подключаться к веб-интерфейсу (необязательно). Например: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Сообщение для информирования пользователей о техническом обслуживании (необязательно). Если параметр не определен, будет использоваться сообщение по умолчанию: *«Zabbix находится на обслуживании»*.

Отображение

Пользователи увидят следующий экран при попытке доступа к веб-интерфейсу Zabbix в режиме обслуживания. Экран обновляется каждые 30 секунд, чтобы после завершения обслуживания автоматически вернуться в нормальное состояние без вмешательства пользователя.



IP-адреса, указанные в *ZBX_GUI_ACCESS_IP_RANGE*, смогут получать доступ к веб-интерфейсу как обычно.

Параметры страницы

Обзор

Большинство страниц веб-интерфейса Zabbix поддерживают различные параметры HTTP GET, которые управляют тем, что будет отображаться. Их можно передавать, указывая пары параметр=значение после URL, отделяя их от URL вопросительным знаком (?) и друг от друга амперсандом (&).

Мониторинг > Проблемы

Поддерживаются следующие параметры:

Parameter	Description	Example
show	Параметр фильтра *Показать*. Возможные значения: 1 - недавние проблемы; 2 - все; 3 - в состоянии проблемы.	show=1
name	Параметр фильтра *Проблема*: строка произвольного формата.	name=Zabbix agent
severities	Параметр фильтра *Серьезность*: массив выбранных уровней серьезности в формате `severities[*]=*` (замените * на уровень серьезности). Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.	severities[3]=3
inventory	Параметр фильтра *Инвентаризация узла сети*: массив полей инвентаризации `[field],[value]`	inventory[0][field]=type&invento
evaltype	Параметр фильтра *Теги*: метод оценки тегов. Возможные значения: 0 - и/или; 2 - или.	evaltype=0
tags	Параметр фильтра *Теги*: массив заданных тегов `[tag], [operator], [value]` Возможные значения оператора: 0 - содержит; 1 - равно; 2 - не содержит; 3 - не равно; 4 - существует; 5 - не существует.	tags[0][tag]=target&tags[0][ope
show_tags	Параметр фильтра *Показывать теги*. Возможные значения: 0 - нет; 1 - один; 2 - два; 3 - три.	show_tags=3
tag_name_format	Параметр фильтра *Имя тега*. Возможные значения: 0 - полное имя; 1 - сокращенное имя; 2 - нет.	tag_name_format=1

Parameter	Description	Example
tag_priority	Параметр фильтра *Приоритет отображения тегов*: строка приоритетов отображения тегов, разделенных запятыми	tag_priority=customer, target
show_suppressed	Параметр фильтра *Показывать подавленные проблемы*. Возможные значения: 1 - показывать; 2 - не показывать.	show_suppressed=1
acknowledgement_status	Параметр фильтра *Статус подтверждения*. Возможные значения: 0 - все; 1 - неподтвержденные; 2 - подтвержденные.	acknowledgement_status=0
acknowledged_by_me	Параметр фильтра *Мною*; поддерживается только с `acknowledgement_status=2`. Возможные значения: 0 - отключено; 1 - включено.	acknowledged_by_me=1
compact_view	Параметр фильтра *Компактный вид*. Возможные значения: 0 - отключено; 1 - включено.	compact_view=1
highlight_row	Параметр фильтра *Выделять всю строку*. Возможные значения: 0 - отключено; 1 - включено.	highlight_row=1
filter_name	Параметр свойств фильтра *Имя*: строка произвольного формата	filter_name=Databases
filter_show_counter	Параметр свойств фильтра *Показывать число записей*. Возможные значения: 0 - отключено; 1 - включено.	filter_show_counter=1
filter_custom_time	Параметр свойств фильтра *Задать пользовательский период времени*. Возможные значения: 0 - отключено; 1 - включено.	filter_custom_time=1
sort	Столбец, по которому выполняется сортировка. Возможные значения: clock - сортировка по столбцу *Время*; host - сортировка по столбцу *Узел сети*; severity - сортировка по столбцу *Серьезность*; name - сортировка по столбцу *Проблема*.	sort=clock
sortorder	Порядок сортировки результатов. Возможные значения: DESC - по убыванию; ASC - по возрастанию.	sortorder=DESC
age_state	Параметр фильтра *Меньше чем возраст*; поддерживается только с `show=3`. Возможные значения: 0 - отключить параметр age; 1 - включить параметр age.	age_state=1
age	Параметр фильтра *Меньше чем возраст*: целое число, количество дней; поддерживается только с `age_state=1` и `show=3`.	age=7
groupids	Параметр фильтра *Группы узлов сети*: массив ID групп узлов сети.	groupids[]=4

Parameter	Description	Example
hostids	Параметр фильтра *Узлы сети*: массив ID узлов сети	hostids[]=10084
triggerids	Параметр фильтра *Триггеры*: массив ID триггеров	triggerids[]=22382
show_timeline	Параметр фильтра *Показывать временную шкалу*; не поддерживается с `compact_view=1`. Возможные значения: 0 - не показывать; 1 - показывать.	show_timeline=1
details	Параметр фильтра *Показывать подробности*. Возможные значения: 0 - не показывать; 1 - показывать.	details=1
from	Начало диапазона дат, может быть относительным (например, now-1m); поддерживается только с `filter_custom_time=1`	from=now-2h
to	Конец диапазона дат, может быть относительным (например, now-1m); поддерживается только с `filter_custom_time=1`	to=now

См. также: параметры фильтра страницы Problems.

Режим киоска

Режим киоска на поддерживаемых страницах веб-интерфейса можно активировать с помощью параметров URL. Например, на панелях мониторинга:

- `/zabbix.php?action=dashboard.view&kiosk=1` - активировать режим киоска
- `/zabbix.php?action=dashboard.view&kiosk=0` - активировать обычный режим

Слайд-шоу

На панели мониторинга можно активировать слайд-шоу:

- `/zabbix.php?action=dashboard.view&slideshow=on` - активировать слайд-шоу

Определения

Обзор

Хотя многие параметры веб-интерфейса можно настроить, используя сам веб-интерфейс, однако, некоторые настройки в настоящее время возможно изменить только через файл определений.

Этим файлом является `defines.inc.php` расположенный в папке /include раздела документов Zabbix HTML.

Параметры

Параметры в этом файле, которые могут представлять интерес для пользователей:

- ZBX_MIN_PERIOD

Минимальный период графика, в секундах. По умолчанию — одна минута.

- GRAPH_YAXIS_SIDE_DEFAULT

Расположение оси Y по умолчанию в простых графиках и значение по умолчанию для раскрывающегося списка при добавлении элементов данных в пользовательские графики. Возможные значения: 0 - слева, 1 - справа.

По умолчанию: 0

- ZBX_SESSION_NAME

Строка, используемая в качестве имени cookie сеанса веб-интерфейса Zabbix.

По умолчанию: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Время жизни в секундах, используемое для инвалидирования кэша данных ответа Vault. Установите 0, чтобы отключить кэширование ответов Vault.

По умолчанию: 60

- SUBFILTER_VALUES_PER_GROUP

Количество значений подфильтра на группу (например, в подфильтре последних данных).

По умолчанию: 1000

- ZBX_MAX_WIDGET_LINES

Максимальное количество строк виджета для отображения.

По умолчанию: 1000

Дополнительные языки веб-интерфейса

Обзор Чтобы использовать в веб-интерфейсе Zabbix любой язык, кроме английского, на веб-сервере должна быть установлена соответствующая локаль. Кроме того, для работы переводов требуется расширение PHP gettext.

Установка локалей Чтобы вывести список всех установленных языков, выполните:

```
locale -a
```

Если некоторые необходимые языки отсутствуют в списке, откройте файл `/etc/locale.gen` и раскомментируйте нужные локали. Поскольку Zabbix использует кодировку UTF-8, необходимо выбрать локали с набором символов UTF-8.

Теперь выполните:

```
locale-gen
```

Перезапустите веб-сервер. Обратите внимание, что если PHP-FPM установлен и используется веб-сервером, его также необходимо перезапустить вместе с веб-сервером.

Теперь локали должны быть установлены. Может потребоваться обновить страницу веб-интерфейса Zabbix в браузере с помощью `Ctrl` + `F5`, чтобы появились новые языки.

Установка Zabbix Если Zabbix устанавливается напрямую из репозитория Zabbix git, файлы переводов следует создавать вручную. Чтобы создать файлы переводов, выполните:

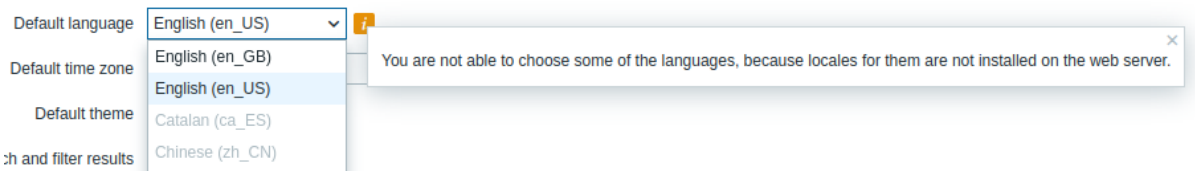
```
make gettext
locale/make_mo.sh
```

Этот шаг не требуется при установке Zabbix из пакетов или исходных файлов `tar.gz`.

Выбор языка В веб-интерфейсе Zabbix есть несколько способов выбрать язык:

- При установке веб-интерфейса — в мастере установки веб-интерфейса. Выбранный язык будет установлен как системный по умолчанию.
- После установки системный язык по умолчанию можно изменить в разделе *Administration* > *General* > *GUI*.
- Язык для конкретного пользователя можно изменить в профиле пользователя.

Если для языка на машине не установлен соответствующий locale, этот язык будет отображаться серым в селекторе языков Zabbix. Если отсутствует хотя бы один locale, рядом с селектором языков отображается оранжевый значок. При нажатии на этот значок будет показано следующее сообщение: "You are not able to choose some of the languages, because locales for them are not installed on the web server."



Создание собственной темы

Обзор

По умолчанию Zabbix предоставляет ряд предопределенных тем веб-интерфейса. Чтобы создать собственную тему, следуйте пошаговой процедуре, приведенной здесь.

Шаг 1: Создайте свою тему

Чтобы создать собственную тему:

1. Создайте CSS-файл с нуля или скопируйте существующую тему (например, `blue-theme.css`, `dark-theme.css` или другую) из каталога `assets/styles/` и измените ее.
2. Сохраните свою тему (например, `custom-theme.css`) в каталоге `assets/styles/`.

Шаг 2: Включение темы в списке тем

Чтобы сделать пользовательскую тему доступной в веб-интерфейсе, необходимо добавить ее в список тем, возвращаемый методом `APP::getThemes()`. Для этого переопределите метод `ZBase::getThemes()` внутри класса `APP` следующим образом:

1. Откройте файл `include/classes/core/APP.php` и найдите класс `APP`:

```
class APP extends ZBase {

}
```

2. Внутри класса перед его последней закрывающей фигурной скобкой } добавьте следующий метод:

```
public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme')
          ]);
}
```

где:

- `custom-theme` - внутренний идентификатор темы, который должен совпадать с именем CSS-файла без расширения `.css`;
- `Custom theme` - отображаемое имя, которое будет видно в веб-интерфейсе.

Чтобы добавить несколько тем, перечислите их в массиве, разделяя каждую запись запятой. Запятая после последней записи является необязательной.

Например:

```
      public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme'),
              'anothertheme' => _('Another theme'),
              'onemoretheme' => _('One more theme')
          ]);
      }
```

Шаг 3: Изменение цветов графиков

Чтобы изменить цвета графиков, необходимо добавить запись для новой темы в таблицу базы данных `graph_theme`.

Пример добавления записи в базу данных MySQL/MariaDB:

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

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
```

```
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

Шаг 4. Использование новой темы

В веб-интерфейсе Zabbix можно установить эту тему в качестве темы по умолчанию или выбрать ее в профиле пользователя.

Режим отладки

Обзор

Режим отладки можно использовать для диагностики проблем с производительностью страниц веб-интерфейса Zabbix.

Конфигурация

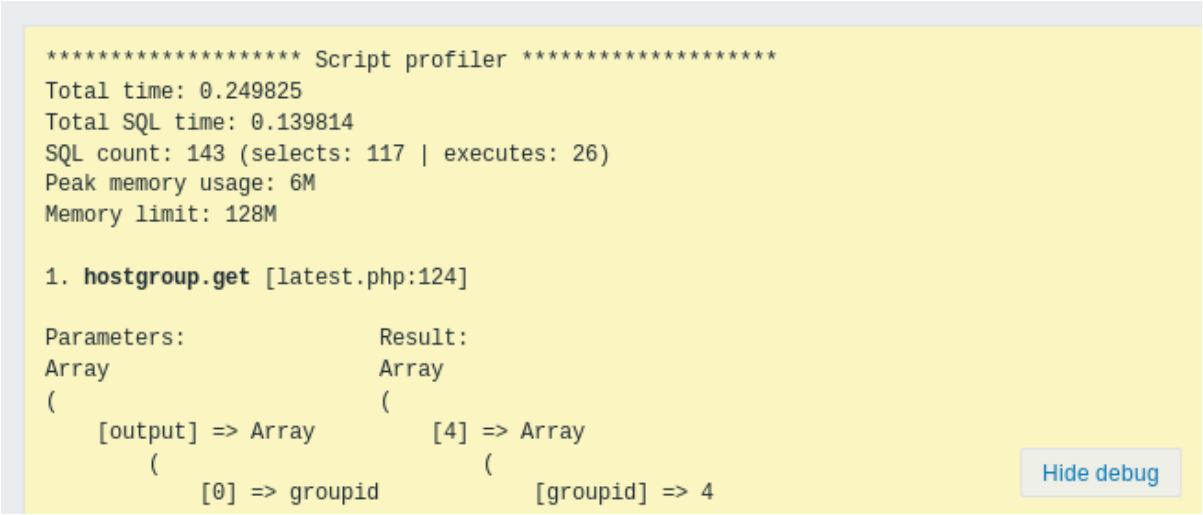
Режим отладки можно включить для отдельных пользователей, которые входят в группу пользователей:

- при настройке группы пользователей;
- при просмотре настроенных групп пользователей.

Когда для группы пользователей включен *Режим отладки*, ее пользователи увидят кнопку *Debug* в правом нижнем углу окна браузера:



При нажатии кнопки *Debug* под содержимым страницы открывается новое окно, в котором отображается SQL-статистика страницы, а также список вызовов API и отдельных SQL-операторов:



В случае проблем с производительностью страницы это окно можно использовать для поиска первопричины проблемы.

Warning:
Включенный *Режим отладки* негативно влияет на производительность веб-интерфейса.

Файлы cookie, используемые Zabbix

Обзор

На этой странице приведен список cookie, используемых Zabbix.

Name	Description	Values	Expires/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Данные сеанса веб-интерфейса Zabbix, хранящиеся в виде JSON, закодированного base64		Сеанс (истекает при завершении сеанса просмотра)	+	+ (только если на веб-сервере включен HTTPS)
tab	Номер активной вкладки; этот cookie используется только на страницах с несколькими вкладками (например, на странице настройки *Host*, *Trigger* или *Action*) и создается, когда пользователь переходит с основной вкладки на другую вкладку (например, *Tags* или *Dependencies*). 0 используется для основной вкладки.	Пример: 1	Сеанс (истекает при завершении сеанса просмотра)	-	-
browserwarning_ignore	Сохранить или игнорировать предупреждение об использовании устаревшего браузера.	yes	Сеанс (истекает при завершении сеанса просмотра)	-	-
system-message-ok	Сообщение, которое будет показано сразу после перезагрузки страницы.	Сообщение в виде обычного текста	Сеанс (истекает при завершении сеанса просмотра) или сразу после перезагрузки страницы	+	-

Name	Description	Values	Expires/Max-Age	HttpOnly[1]	Secure[2]
system-message-error	Сообщение об ошибке, которое будет показано сразу после перезагрузки страницы.	Сообщение в виде обычного текста	Сеанс (истекает при завершении сеанса просмотра) или сразу после перезагрузки страницы	+	-

Note:
Принудительное включение флага 'HttpOnly' для cookie Zabbix с помощью директивы веб-сервера не поддерживается.

Примечания

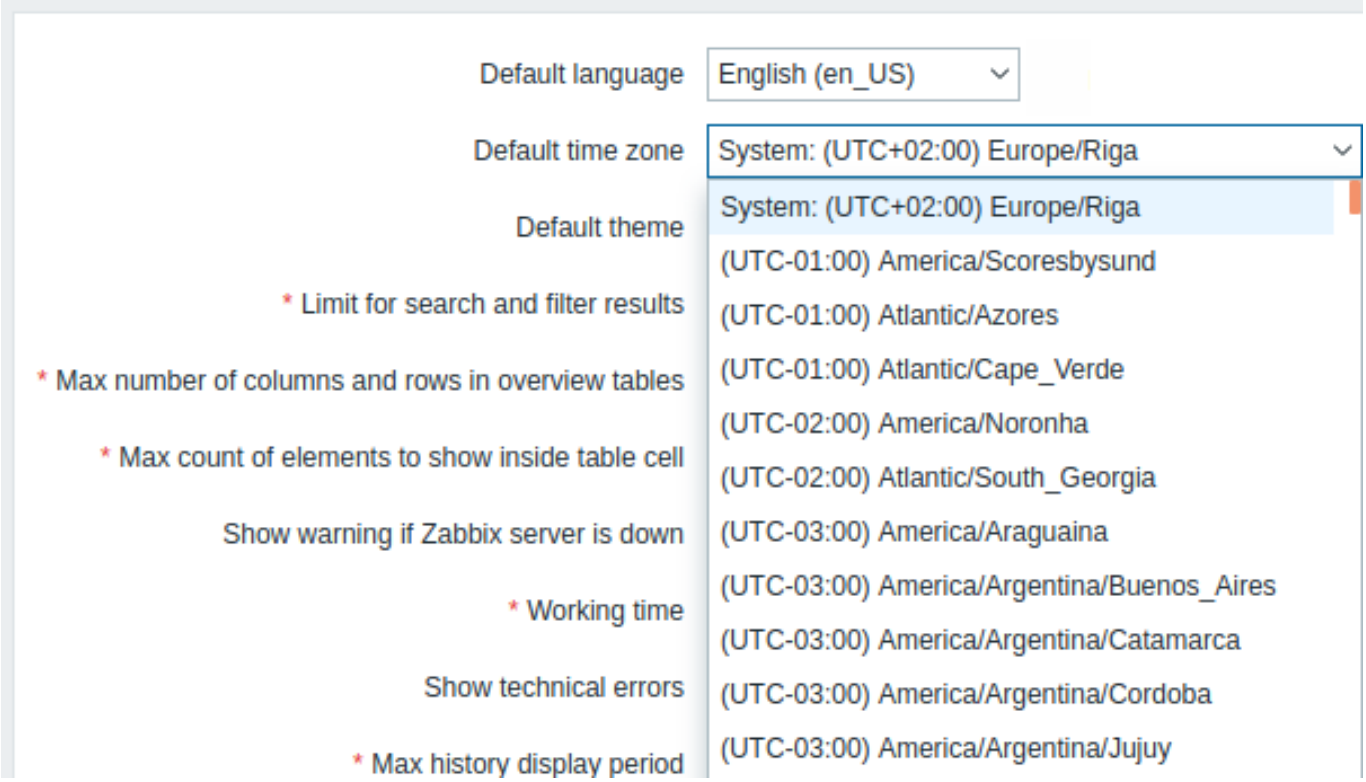
[1] Если `HttpOnly` имеет значение 'true', cookie будет доступен только через протокол HTTP. Это означает, что cookie не будет доступен для скриптовых языков, таких как JavaScript. Эта настройка может эффективно помочь снизить риск кражи идентификационных данных через атаки XSS (хотя она поддерживается не всеми браузерами).

[2] `Secure` означает, что cookie должен передаваться только по защищенному HTTPS-соединению от клиента. Если установлено значение 'true', cookie будет установлен только при наличии защищенного соединения.

Часовые пояса

Обзор

Часовой пояс веб-интерфейса можно задать глобально в веб-интерфейсе и настроить для отдельных пользователей.



Если выбран вариант *System*, для веб-интерфейса будет использоваться часовой пояс веб-сервера (включая значение `date.timezone` в `php.ini`, если оно задано), а сервер Zabbix будет использовать часовой пояс машины, на которой он запущен.

Note:
Сервер Zabbix будет использовать указанный глобальный часовой пояс или часовой пояс пользователя только при раскрытии макросов в уведомлениях (например, `{EVENT.TIME}` может раскрываться с учетом разных часовых поясов для разных пользователей) и для ограничения времени отправки уведомлений (см. настройку «Когда активно» в настройке пользовательских средств оповещения).

Выбор часового пояса не влияет на формат даты и времени в веб-интерфейсе. Вместо этого можно изменить язык интерфейса (при установке или в разделе настроек пользователя) - выбор *English (en_US)* также включит американский формат даты и времени в веб-интерфейсе.

Конфигурация

Глобальный часовой пояс:

- Может быть задан вручную при установке веб-интерфейса.
- Может быть изменен в *Администрирование* > *Общее* > *GUI*.

Часовой пояс на уровне пользователя:

- Может быть задан при настройке/обновлении пользователя.
- Может быть задан каждым пользователем в его профиле пользователя.

См. также: Согласование часовых поясов при использовании интервалов планирования.

Сброс пароля

Обзор В этом разделе описаны шаги по сбросу паролей пользователей в Zabbix.

Шаги Обратитесь к администратору Zabbix, если вы забыли пароль Zabbix и не можете войти в систему.

Пользователь с ролью суперадминистратора может изменять пароли всех пользователей в форме настройки пользователей.

Если суперадминистратор забыл пароль и не может войти в систему, необходимо выполнить следующий SQL-запрос, чтобы установить пароль по умолчанию для пользователя-суперадминистратора (замените `Admin` на соответствующее имя пользователя-суперадминистратора):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

После выполнения этого запроса паролем пользователя будет установлено значение *zabbix*. Обязательно измените пароль по умолчанию при первом входе в систему.

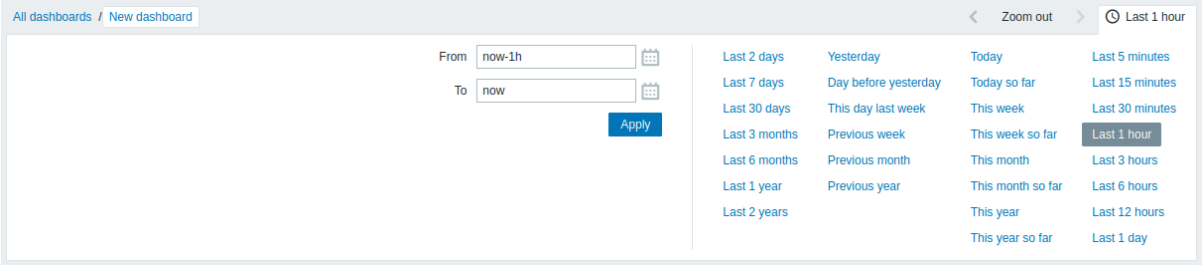
Период времени и селекторы узлов сети

Обзор

На этой странице описаны доступные на панелях мониторинга селекторы периода времени и узлов сети. Эти селекторы позволяют динамически изменять отображаемые данные во всех совместимых виджетах панелей мониторинга без изменения настроек отдельных виджетов.

Выбор периода времени

С помощью выбора периода времени можно выбрать часто используемые периоды одним щелчком мыши. Его можно развернуть или свернуть, нажав кнопку выбора периода времени в правом верхнем углу панели мониторинга.



Такие варианты, как *Сегодня*, *На этой неделе* и т. д., отображают весь период, включая часы или дни в будущем. Такие варианты, как *Сегодня на данный момент*, *На этой неделе на данный момент* и т. д., отображают только прошедшие часы.

После выбора периода его можно перемещать вперед и назад во времени, нажимая кнопки со стрелками . Кнопка *Уменьшить масштаб* позволяет уменьшить период на 50% в каждом направлении.

Note:
Для графиков (кроме графиков с *Пользовательским* периодом времени) отображаемый период времени также можно выбрать, выделив область на графике левой кнопкой мыши. После отпускания левой кнопки мыши график увеличит масштаб выделенной области. Уменьшить масштаб также можно двойным щелчком на графике.

Поля *От/До* отображают выбранный период либо в формате абсолютного времени (в формате `Y-m-d H:i:s`), либо в формате относительного времени. Относительный период времени может содержать одну или несколько математических операций (- или +), например `now-1d` или `now-1d-2h+5m`.

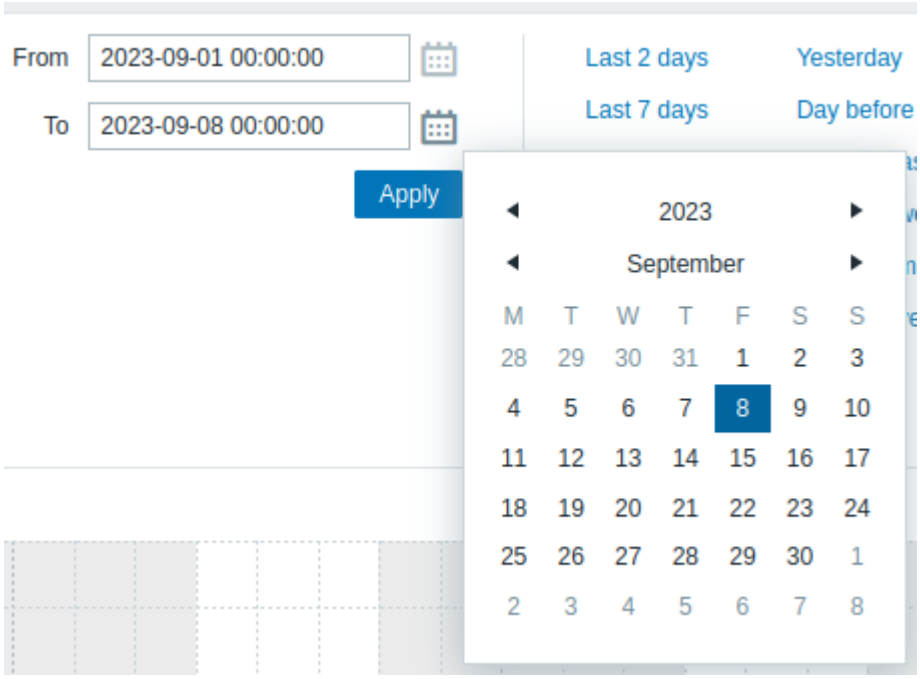
Поддерживаются следующие сокращения для относительного времени:

- `now`
- `s` (секунды)
- `m` (минуты)
- `h` (часы)
- `d` (дни)
- `w` (недели)
- `M` (месяцы)
- `y` (годы)

В выборе периода времени поддерживается точность (например, `/M` в `now-1d/M`). Подробные сведения о точности:

Точность	От	До
`m`	Y-m-d H:m:00	Y-m-d H:m:59
`h`	Y-m-d H:00:00	Y-m-d H:59:59
`d`	Y-m-d 00:00:00	Y-m-d 23:59:59
`w`	Понедельник недели 00:00:00	Воскресенье недели 23:59:59
`M`	Первый день месяца 00:00:00	Последний день месяца 23:59:59
`y`	1 января года 00:00:00	31 декабря года 23:59:59

Также можно выбрать период времени с помощью средства выбора даты. Чтобы открыть его, нажмите значок календаря рядом с полями *От/До*.



Note:
В средстве выбора даты можно перемещаться между годом, месяцем и датой с помощью клавиш `Tab`, `Shift+Tab` и клавиш со стрелками. Нажмите `Enter`, чтобы подтвердить выбор.

Примеры:

From	To	Selected period
now/d	now/d	00:00 - 23:59 сегодня
now/d	now/d+1d	00:00 сегодня - 23:59 завтра
now/w	now/w	Понедельник 00:00:00 - воскресенье 23:59:59 на этой неделе
now-1y/w	now-1y/w	Неделя с понедельника 00:00:00 - воскресенья 23:59:59 год назад

Attention:
Использование "now/M+1M" для параметра *To* может добавить 31 день, что может привести к смещению даты на 1-3 дня в зависимости от количества дней в месяце. Например, если использовать это в январе, результатом может быть 02 марта вместо ожидаемого 28 февраля. Чтобы избежать этой проблемы, используйте "now/M-3d+1M/M", который корректно учитывает длину месяца. Аналогично, если настраиваете *From* для перехода назад, используйте "now/M+3d-1M/M".

Выбор узла сети

С помощью выбора узла сети можно выбрать узел сети, данные которого будут отображаться в совместимых виджетах панели мониторинга.

Он становится доступен после редактирования и сохранения панели мониторинга, если хотя бы один виджет настроен с параметром *Переопределить узел сети*, для которого в качестве источника данных выбрана панель мониторинга.



Параметры конфигурации: веб-интерфейс Zabbix

Обзор Параметры конфигурации позволяют настраивать поведение веб-интерфейса Zabbix.

Укажите значения параметров в файле конфигурации веб-интерфейса (`zabbix.conf.php`).

Эта страница состоит из следующих разделов:

- Обзор параметров (чтобы просмотреть подробные сведения о параметре, нажмите его имя)
- Подробные сведения о параметрах

Параметр	Описание
$DB['TYPE']	Тип базы данных.
$DB['SERVER']	Имя узла сети базы данных или IP-адрес.
$DB['PORT']	Порт базы данных.
$DB['DATABASE']	Имя базы данных, используемой веб-интерфейсом Zabbix для подключения к настроенной базе данных.
$DB['USER']	Имя пользователя, используемое веб-интерфейсом Zabbix для подключения к базе данных.
$DB['PASSWORD']	Пароль, используемый веб-интерфейсом Zabbix для подключения к базе данных.
$DB['SCHEMA']	Имя схемы базы данных, используемой для PostgreSQL.
$DB['ENCRYPTION']	Включение или отключение шифрования TLS для соединений между веб-интерфейсом Zabbix и базой данных.
$DB['KEY_FILE']	Путь к файлу закрытого ключа клиента, используемого для TLS-соединения с базой данных.
$DB['CERT_FILE']	Путь к файлу сертификата клиента, используемого для TLS-соединения с базой данных.
$DB['CA_FILE']	Путь к файлу сертификата центра сертификации (CA), используемого для TLS-соединения с базой данных.
$DB['VERIFY_HOST']	Включение или отключение проверки имени узла сети сервера базы данных в сертификате TLS.
$DB['CIPHER_LIST']	Список разрешенных шифров TLS для подключения к базе данных.
$DB['VAULT']	Укажите менеджер секретов Vault, используемый для хранения учетных данных базы данных.
$DB['VAULT_URL']	URL менеджера секретов Vault, используемого для получения учетных данных базы данных.

Параметр	Описание
$DB['VAULT_PREFIX']	Префикс пути или запроса Vault, используемого для получения учетных данных базы данных.
$DB['VAULT_DB_PATH']	Путь к учетным данным базы данных в менеджере секретов Vault.
$DB['VAULT_TOKEN']	Токен аутентификации, используемый для доступа к менеджеру секретов Vault.
$DB['VAULT_APP_ROLE_ID']	Идентификатор роли AppRole, используемый для аутентификации в HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	Секретный идентификатор AppRole, используемый для аутентификации в HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Путь к файлу сертификата клиента, используемого для аутентификации в менеджере секретов Vault.
$DB['VAULT_KEY_FILE']	Путь к файлу закрытого ключа клиента, используемого для аутентификации в менеджере секретов Vault.
$DB['VAULT_CACHE']	Включение или отключение кэширования учетных данных, полученных из менеджера секретов Vault.
$ZBX_SERVER	Укажите значение для переопределения имени узла сети/IP-адреса Zabbix.
$ZBX_SERVER_PORT	Укажите значение для переопределения порта сервера Zabbix.
$ZBX_SERVER_NAME	Имя сервера Zabbix.
$IMAGE_FORMAT_DEFAULT	Формат изображения по умолчанию.
$HISTORY_PROVIDERS[]	Настройка поставщиков хранения истории для Elasticsearch или ClickHouse.
$SSO['SETTINGS']	Параметры конфигурации для входа с использованием SAML.
$SSO['CERT_STORAGE']	Укажите место хранения сертификатов SAML.
$SSO['SP_KEY']	Путь к файлу закрытого ключа поставщика услуг SAML (SP), если закрытый ключ хранится в файловой системе.
$SSO['SP_CERT']	Путь к файлу сертификата поставщика услуг SAML (SP), если сертификат хранится в файловой системе.
$SSO['IDP_CERT']	Путь к файлу сертификата поставщика удостоверений SAML, если сертификат хранится в файловой системе.
$ZBX_FEATURE_FLAGS['banner_enabled']	Включение или отключение поддержки баннеров.
$ZBX_FEATURE_FLAGS['http_auth_enabled']	Включение или отключение HTTP-аутентификации пользователей.
$ZBX_FEATURE_FLAGS['module_config_enabled']	Включение или отключение доступа к модулям.
$ZBX_FEATURE_FLAGS['media_type_deprecated']	Запрет редактирования определенных предварительно настроенных типов носителей.
$ZBX_SERVER_TLS['ACTIVE']	Указывает, включен ли TLS для соединений с сервером Zabbix.
$ZBX_SERVER_TLS['CA_FILE']	Путь к файлу сертификата центра сертификации (CA), используемого для проверки сертификата сервера.
$ZBX_SERVER_TLS['KEY_FILE']	Путь к файлу закрытого ключа клиента, соответствующего сертификату клиента.
$ZBX_SERVER_TLS['CERT_FILE']	Путь к файлу сертификата клиента, если требуется взаимная аутентификация TLS.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Различающееся имя (DN) издателя для сопоставления с сертификатом сервера.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Различающееся имя (DN) субъекта для сопоставления с сертификатом сервера.
$TELEMETRY_PROVIDERS[]	Настройка источника данных телеметрии.

Сведения о параметрах

$DB['TYPE']

Тип базы данных.

Значения: MYSQL - MySQL, POSTGRESQL - PostgreSQL.

$DB['SERVER']

Имя узла сети базы данных или IP-адрес.

Если $DB['SERVER'] не определён, используется сокет домена Unix по умолчанию. Для MySQL установка $DB['SERVER'] в значение localhost также приводит к подключению через сокет Unix по умолчанию.
 Для PostgreSQL $DB['SERVER'] может содержать несколько записей host:port, разделённых запятыми. Если для узла сети в списке, разделённом запятыми, порт не указан, в качестве порта по умолчанию используется значение, заданное в $DB['PORT']. Если в $DB['SERVER'] указан путь к сокету (например, /var/run/pgbouncer), используется этот сокет домена Unix.

Примеры:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Порт базы данных.

Если значение не задано, используется значение 0.

$DB['DATABASE']

Имя базы данных, используемое веб-интерфейсом Zabbix для подключения к настроенной базе данных.

Пример:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Имя пользователя, используемое веб-интерфейсом Zabbix для подключения к базе данных.

$DB['PASSWORD']

Пароль, используемый веб-интерфейсом Zabbix для подключения к базе данных.

$DB['SCHEMA']

Имя схемы базы данных, используемое для PostgreSQL.

$DB['ENCRYPTION']

Включение или отключение шифрования соединений между веб-интерфейсом Zabbix и базой данных.

Значения: `true` — включено, `false` — отключено.

$DB['KEY_FILE']

Путь к файлу закрытого ключа клиента, используемого для TLS-соединения с базой данных.

Пример:

```
//    MySQL:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Путь к файлу сертификата клиента, используемого для TLS-соединения с базой данных.

Пример:

```
//    MySQL:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

Путь к файлу сертификата CA, используемому для TLS-соединения с базой данных.

Примеры:

```
//    MySQL:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
//    PostgreSQL:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Включение или отключение проверки имени узла сети сервера базы данных в сертификате TLS.

Значения: `true` - включено, `false` - отключено.

$DB['CIPHER_LIST']

Список разрешенных шифров TLS для подключения к базе данных.

Примеры:

```
//    MySQL:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Укажите менеджер секретов Vault, используемый для хранения учетных данных базы данных.

Допустимые значения: `CyberArk` или `HashiCorp`.

Примеры:

```
//    CyberArk:
$DB['VAULT'] = 'CyberArk';
//    HashiCorp:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

URL диспетчера секретов Vault, используемого для получения учетных данных базы данных.

Примеры:

```
//    CyberArk:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
//    HashiCorp:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Префикс пути или запроса Vault, используемый для получения учетных данных базы данных.

Пример:

```
//    HashiCorp:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Путь к учетным данным базы данных в менеджере секретов Vault.

Примеры:

```
//    CyberArk:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
//    HashiCorp:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Токен аутентификации, используемый для доступа к диспетчеру секретов Vault.

Пример:

```
//    HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

Идентификатор роли VaultAppSecretID, используемый для аутентификации в HashiCorp Vault.

Пример:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID используется для аутентификации в HashiCorp Vault.

Пример:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Путь к файлу сертификата клиента, используемого для аутентификации в менеджере секретов Vault.

Пример:

```
//    CyberArk:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Путь к файлу закрытого ключа клиента, используемого для аутентификации в диспетчере секретов Vault.

Пример:

```
//    CyberArk:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Включает или отключает кэширование учетных данных, полученных из менеджера секретов Vault.

Значения: `true` — включено, `false` — отключено.

$ZBX_SERVER

Укажите значение для переопределения имени узла сети или IP-адреса Zabbix.
 Если $ZBX_SERVER указан без $ZBX_SERVER_PORT, используется порт по умолчанию (10051).

$ZBX_SERVER_PORT

Укажите значение для переопределения порта сервера Zabbix.
 Этот параметр используется вместе с $ZBX_SERVER.

$ZBX_SERVER_NAME

Имя сервера Zabbix.

$IMAGE_FORMAT_DEFAULT

Формат изображения по умолчанию.

По умолчанию: IMAGE_FORMAT_PNG
 Значения: - IMAGE_FORMAT_PNG - формат PNG - IMAGE_FORMAT_JPEG - формат JPEG - IMAGE_FORMAT_GIF - формат GIF

$HISTORY_PROVIDERS[]

Настройка поставщиков хранения истории для Elasticsearch или ClickHouse.

Поддерживаемые параметры конфигурации: - types - массив типов данных, которые будут храниться во внешнем хранилище; - provider - тип поставщика истории: elasticsearch или clickhouse; - url - URL поставщика истории; - db - имя базы данных (используется для ClickHouse); - username - пользователь базы данных (используется для ClickHouse); - password - пароль базы данных (используется для ClickHouse).

Примеры:

```
//    ClickHouse:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
//    Elasticsearch:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

Параметры конфигурации для входа через SAML. Массив $SSO['SETTINGS'] должен соответствовать структуре, ожидаемой библиотекой *SAML PHP Toolkit* (поставляется вместе с Zabbix).

В $SSO['SETTINGS'] можно указать только следующие параметры:

- strict
- baseurl
- compress
- contactPerson
- organization
- sp (только параметры, указанные в этом списке)
  - attributeConsumingService
  - x509certNew
- idp (только параметры, указанные в этом списке)
  - singleLogoutService (только один параметр)

```
        * responseUrl
    - certFingerprint
    - certFingerprintAlgorithm
    - x509certMulti
```

- security (только параметры, указанные в этом списке)

```
    - signMetadata
    - wantNameId
    - requestedAuthnContext
    - requestedAuthnContextComparison
    - wantXMLValidation
    - relaxDestinationValidation
    - destinationStrictlyMatches
    - rejectUnsolicitedResponsesWithInResponseTo
    - signatureAlgorithm
    - digestAlgorithm
    - lowercaseUrlencoding
```

Все остальные параметры берутся из базы данных, и их нельзя переопределить. Параметр debug будет проигнорирован.

Если веб-интерфейс Zabbix находится за прокси или балансировщиком нагрузки, можно использовать пользовательский параметр use_proxy_headers:

- false (по умолчанию) - игнорировать HTTP-заголовки X-Forwarded-*;
- true - использовать HTTP-заголовки X-Forwarded-* для формирования базового URL.

Полное описание доступных параметров конфигурации см. в официальной документации библиотеки.

Пример:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Укажите, где хранятся сертификаты SAML.

По умолчанию: file
 Значения: - database - закрытый ключ и сертификаты хранятся в базе данных; - file - закрытый ключ и сертификаты хранятся в файловой системе.

Пример:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Путь к файлу закрытого ключа поставщика услуг SAML (SP), если закрытый ключ хранится в файловой системе.

Пример:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Путь к файлу сертификата поставщика услуг SAML (SP), если сертификат хранится в файловой системе.

Пример:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Путь к файлу сертификата поставщика удостоверений SAML, если сертификат хранится в файловой системе.

Пример:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Включение или отключение поддержки баннеров.

По умолчанию: `true<br>` Значения: `true` - включено, `false` - отключено.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Включение или отключение HTTP-аутентификации пользователей.

По умолчанию: `true<br>` Значения: `true` — включено, `false` — отключено.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Включение или отключение доступа к модулям.

По умолчанию: `true<br>` Значения: `true` — включено, `false` — отключено.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Запрет редактирования определенных предварительно настроенных типов носителей.

Значения: `email`, `script`, `sms`, `webhook`, `push`.

Пример:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Указывает, включено ли TLS-соединение между веб-интерфейсом Zabbix и сервером Zabbix.

Значение по умолчанию: `false<br>` Значения: `true` - включено, `false` - отключено.

$ZBX_SERVER_TLS['CA_FILE']

Путь к файлу сертификата центра сертификации (CA), используемого для проверки сертификата сервера.

Пример:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Путь к файлу закрытого ключа клиента, соответствующего сертификату клиента, используемому для взаимной аутентификации TLS.

Пример:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Путь к файлу сертификата клиента, если требуется взаимная аутентификация TLS.

Пример:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Distinguished Name (DN) издателя для сопоставления с сертификатом сервера.

Пример:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Distinguished Name (DN) субъекта, с которым должно совпадать значение в сертификате сервера.

Пример:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Конфигурация источника данных телеметрии.

Поддерживаемые параметры конфигурации для ClickHouse: - `url` - задает URL HTTP или HTTPS, используемый для подключения к источнику данных телеметрии. - db - задает имя базы данных. - `username` - задает имя пользователя,

используемое для входа в базу данных. - password - задает пароль, используемый для входа в базу данных. - ssl_cert_file, ssl_key_file, ssl_key_password, ssl_verify_peer, ssl_verify_host, ssl_ca_location, ssl_ca_file - задают параметры для настройки защищенного (TLS/SSL) подключения к базе данных. - vault_path - задает путь к секрету Vault, содержащему значения username и password. Не используйте вместе с параметрами username и password.

Пример:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Обзор Zabbix API позволяет программно получать и изменять конфигурацию Zabbix, а также предоставляет доступ к историческим данным. Она широко используется для:

- Создания новых приложений для работы с Zabbix.
- Интеграции Zabbix со сторонним программным обеспечением.
- Автоматизации рутинных задач.

Zabbix API — это API на основе HTTP, и оно поставляется как часть веб-интерфейса. Оно использует протокол JSON-RPC 2.0, что означает две вещи:

- API состоит из набора отдельных методов.
- Запросы и ответы между клиентами и API кодируются с использованием формата JSON.

Дополнительные сведения о протоколе и JSON см. в спецификации JSON-RPC 2.0 и на главной странице формата JSON.

Дополнительные сведения об интеграции функциональности Zabbix в ваши приложения Python см. в библиотеке Python для Zabbix.

> **Note:**
> Доступ пользователей в Zabbix, включая как конфигурацию, так и исторические данные, зависит от типа пользователя, назначенной роли пользователя и групп пользователей.

Структура API состоит из ряда методов, которые формально сгруппированы в отдельные API. Каждый из методов выполняет одну конкретную задачу. Например, метод host.create относится к *host* API и используется для создания новых узлов сети. Исторически API иногда называют "классами".

Большинство API содержат как минимум четыре метода: get, create, update и delete для получения, создания, обновления и удаления данных соответственно, однако некоторые API могут предоставлять совершенно иной набор методов.

Выполнение запросов После настройки веб-интерфейса вы можете использовать удаленные HTTP-запросы для вызова API. Для этого необходимо отправлять HTTP POST-запросы к файлу api_jsonrpc.php, расположенному в каталоге веб-интерфейса.
 Например, если ваш веб-интерфейс Zabbix установлен в https://example.com/zabbix, HTTP-запрос для вызова метода apiinfo.version может выглядеть так:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

Запрос должен содержать заголовок Content-Type, установленный в одно из следующих значений: application/json-rpc, application/json или application/jsonrequest.

Объект запроса должен содержать следующие свойства:

- jsonrpc - версия протокола JSON-RPC, используемого API (Zabbix API реализует JSON-RPC версии 2.0);

- `method` - вызываемый метод API;
- `params` - параметры, которые будут переданы методу API;
- `id` - произвольный идентификатор запроса (если не указан, API рассматривает запрос как уведомление).

Если запрос корректен, ответ, возвращаемый API, должен выглядеть так:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Объект ответа, в свою очередь, содержит следующие свойства:

- `jsonrpc` - версия протокола JSON-RPC;
- `result` - данные, возвращенные методом;
- `id` - идентификатор соответствующего запроса.

Аутентификация Чтобы получить доступ к любым данным в Zabbix, необходимо либо:

- Использовать существующий API token, созданный в веб-интерфейсе Zabbix, в разделе *Users* > *API tokens*, либо созданный с помощью Token API.
- Использовать токен аутентификации, полученный с помощью метода `user.login`.

Например, если вы хотите получить новый токен аутентификации, выполнив вход как обычный пользователь *Admin*, JSON-запрос будет выглядеть так:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

Если учетные данные указаны верно, ответ, возвращаемый API, должен содержать токен аутентификации пользователя:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Методы авторизации Через заголовок "Authorization"

Все запросы к API требуют аутентификации или API-токена. Вы можете передать учетные данные, используя заголовок Authorization в запросе:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
Если у вас возникают проблемы с аутентификацией, см. передачу заголовка Authorization.

API Zabbix принимает заголовки без учета регистра (например, `authorization`, `Authorization` и `AUTHORIZATION` обрабатываются одинаково).

Заголовок Authorization поддерживается в кросс-доменных запросах (CORS).

По cookie-файлам Zabbix

Cookie *"zbx_session"* используется для авторизации запроса API из веб-интерфейса Zabbix, выполняемого при помощи JavaScript (из модуля или пользовательского виджета).

Пример рабочего процесса В следующем разделе более подробно рассматриваются несколько примеров использования.

Получение узлов сети Теперь у вас есть действительный токен аутентификации пользователя (представленный в следующих примерах как переменная), который можно использовать для доступа к данным в Zabbix. Например, вы можете использовать метод `host.get` для получения идентификаторов, имен узлов сети и интерфейсов всех настроенных узлов сети:

Запрос:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data @data.json
```

Note:
`data.json` — это файл, содержащий JSON-запрос. Вместо файла вы можете передать запрос в аргументе `--data`.

`data.json`

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

Объект ответа будет содержать запрошенные данные об узлах сети:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
По соображениям производительности всегда рекомендуется перечислять свойства объекта, которые вы хотите получить. Таким образом вы избежите получения всех данных.

Создание нового элемента данных Теперь создайте новый элемент данных на узле сети "Zabbix server", используя данные, полученные в предыдущем запросе `host.get`. Это можно сделать с помощью метода `item.create`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Успешный ответ будет содержать ID только что созданного элемента данных, который можно использовать для ссылки на элемент данных в следующих запросах:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Метод `item.create`, как и другие *методы создания*, также может принимать массивы объектов и создавать несколько элементов данных за один вызов API.

Создание нескольких триггеров Таким образом, если *методы создания* принимают массивы, вы можете добавить несколько триггеров, например, таких:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

Успешный ответ будет содержать идентификаторы вновь созданных триггеров:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Обновление элемента данных Включите элемент данных, установив для его статуса значение 0:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

Успешный ответ будет содержать ID обновленного элемента данных:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
Метод `item.update`, как и другие *методы обновления*, также может принимать массивы объектов и обновлять несколько элементов данных одним вызовом API.

Обновление нескольких триггеров Включите несколько триггеров, установив для их статуса значение 0:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

Успешный ответ будет содержать идентификаторы обновленных триггеров:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Обработка ошибок До настоящего момента все, что вы пробовали, работало без проблем. Но что произойдет, если вы попробуете выполнить некорректный вызов API? Попробуйте создать еще один узел сети, вызвав `host.create`, но не указывая обязательный параметр `groups`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

В ответе будет сообщение об ошибке:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Если произошла ошибка, вместо свойства `result` объект ответа будет содержать свойство `error` со следующими данными:

- `code` - код ошибки;
- `message` - краткое описание ошибки;
- `data` - более подробное сообщение об ошибке.

Ошибки могут возникать в различных случаях, например при использовании некорректных входных значений, при истечении времени сеанса или при попытке доступа к несуществующим объектам. Ваше приложение должно уметь корректно обрабатывать такие ошибки.

Версии API Чтобы упростить версионирование API, начиная с Zabbix 2.0.4 версия API совпадает с версией самого Zabbix. Вы можете использовать метод `apiinfo.version`, чтобы узнать, с какой версией API вы работаете. Это может быть полезно для настройки вашего приложения с учетом возможностей, зависящих от версии.

Zabbix гарантирует обратную совместимость функций в пределах одной основной версии. При внесении несовместимых изменений между основными релизами Zabbix обычно оставляет старые функции как устаревшие в следующем релизе и удаляет их только в релизе после него. Иногда Zabbix может удалять функции между основными релизами без предоставления какой-либо обратной совместимости. Важно никогда не полагаться на устаревшие функции и как можно скорее переходить на более новые альтернативы.

> **Note:**
> Вы можете отслеживать все изменения, внесенные в API, в журнале изменений API.

Дополнительное чтение Теперь у вас достаточно знаний, чтобы начать работать с Zabbix API, однако не останавливайтесь на этом. Для дальнейшего изучения рекомендуем ознакомиться со списком доступных API.

Справочник методов

В этом разделе содержится обзор функций, которые имеются в Zabbix API, который поможет вам найти варианты использования доступных классов и методов.

Мониторинг Zabbix API позволяет получать доступ к истории и другим данным, собранным во время мониторинга.

Панели мониторинга

Управляйте панелями мониторинга и создавайте на их основе запланированные отчеты.

Dashboard API | Template dashboard API | Report API

Кластер высокой доступности

Получить список узлов сервера и их статус.

API кластера высокой доступности

История

Получение исторических значений, собранных процессами мониторинга Zabbix, для отображения или дальнейшей обработки.

History API

Тренды

Получение значений трендов, рассчитанных сервером Zabbix, для отображения или дальнейшей обработки.

Trend API

События

Получение событий, которые сгенерированы триггерами, сетевым обнаружением и другими системами Zabbix для более гибкого управления ситуациями или интеграции со сторонними инструментами.

API события

Проблемы

Получить проблемы в соответствии с заданными параметрами.

Problem API

Карты

Настройте карты, чтобы создавать подробные динамические представления вашей ИТ-инфраструктуры.

Map API

Задачи

Взаимодействие с менеджером задач Zabbix сервера, создание задач и получение ответа.

API задач

APM API Zabbix позволяет получать доступ к данным телеметрии, собранным для мониторинга производительности приложений (APM).

Трассировки

Получение трассировок и составляющих их интервалов.

API трассировок APM | API интервалов APM

Метрики

Получение точек данных метрик.

API метрик APM

Журналы

Получение записей журнала.

APM log API

Сервисы Zabbix API позволяет получать доступ к данным, собранным в ходе мониторинга сервисов.

Соглашение об уровне обслуживания

Определяйте цели уровня обслуживания (SLO), получайте подробную информацию о показателях уровня обслуживания (SLI) о производительности сервиса.

SLA API

Сервисы

Управляйте сервисами для мониторинга уровня обслуживания и получайте подробную информацию об SLA любого сервиса.

API сервисов

Сбор данных API Zabbix позволяет управлять конфигурацией системы мониторинга.

Узлы сети и группы узлов сети

Управляйте группами узлов сети, узлами сети и всем, что с ними связано, включая интерфейсы узлов сети, макросы узлов сети и периоды обслуживания.

Host API | Host group API | User macro API | Value map API | Maintenance API

Элементы данных

Определяет элементы данных для мониторинга.

API элементов данных

Триггеры

Настройте триггеры для получения уведомлений о проблемах в вашей системе.
Управляйте зависимостями триггеров.

API триггеров

Графики

Редактируйте графики или отдельные элементы графика для более наглядного представления собранных данных.

Graph API | Graph item API

Шаблоны и группы шаблонов

Управляйте шаблонами и связывайте их с узлами сети или другими шаблонами.

Template API | Template group API | Value map API

Низкоуровневое обнаружение

Настройте правила низкоуровневого обнаружения, а также прототипы элементов данных, триггеров и графиков для мониторинга динамических сущностей.

LLD rule API | Item prototype API | Trigger prototype API | Graph prototype API | Host prototype API

Complex event processing

Create complex event processing rules.

Complex event processing API

Корреляция событий

Создавайте пользовательские правила корреляции событий.

Correlation API

Обнаружение сети

Управляйте правилами обнаружения на уровне сети, чтобы автоматически находить и контролировать новые узлы сети. Получите полный доступ к информации об обнаруженных службах и узлах сети.

API правила обнаружения | API проверки обнаружения | API обнаруженного узла сети | API обнаруженной службы

Экспорт и импорт

Экспорт и импорт данных конфигурации Zabbix для резервного копирования конфигурации, миграции или массового обновления конфигурации.

Configuration API

Веб-мониторинг

Настройте веб-сценарии для мониторинга ваших веб-приложений и сервисов.

API веб-сценариев

Оповещения API Zabbix позволяет управлять действиями и оповещениями вашей системы мониторинга.

Действия и оповещения

Определяйте действия и операции, чтобы уведомлять пользователей о определенных событиях или автоматически выполнять удаленные команды. Получайте доступ к информации о сгенерированных оповещениях и их получателях.

Action API | Alert API

Типы оповещений

Настройте типы оповещений и несколько способов, которыми пользователи будут получать уведомления.

API типов оповещений

Скрипты

Настраивайте и выполняйте скрипты, чтобы помочь себе в повседневных задачах.

Script API

Пользователи Zabbix API позволяет управлять пользователями вашей системы мониторинга.

Пользователи и группы пользователей

Добавляйте пользователей, которые будут иметь доступ к Zabbix, назначайте их в группы пользователей и предоставляйте разрешения. Создавайте роли для детального управления правами пользователей.

User API | User group API | User directory API | User role API

API-токены

Управляйте токенами авторизации.

Token API

Аутентификация

Измените параметры конфигурации аутентификации.

API аутентификации

Устройства

Управляйте мобильными устройствами для использования с мобильным приложением Zabbix.

API устройств

Администрирование С помощью Zabbix API вы можете изменять настройки администрирования вашей системы мониторинга.

Общие сведения

Изменение некоторых глобальных параметров конфигурации.

Autoregistration API | Icon map API | Image API | Settings API | Regular expression API | Module API | Connector API

Журнал аудита

Отслеживайте изменения конфигурации, внесенные каждым пользователем.

API журнала аудита

Очистка

Настройте очистку.

Housekeeping API

Прокси и группы прокси

Управляйте прокси, используемыми в вашей распределенной системе мониторинга.

Proxy API | Proxy group API

Макросы

Управление макросами.

API пользовательских макросов

Информация о Zabbix Получение версии API Zabbix или времени последнего обновления с сервера Zabbix.

API info API | Server info API

Action

Этот класс предназначен для работы с действиями.

Ссылки на объекты:

- Action
- Action operation
  - Action operation message
  - Action operation condition
- Action recovery operation
- Action update operation
- Action filter
  - Action filter condition

Доступные методы:

- action.create - создать новые действия
- action.delete - удалить действия
- action.get - получить действия
- action.update - обновить действия

Объект действия

Следующие объекты напрямую связаны с методом API `action`.

Действие

Объект действия имеет следующие свойства.

Property	Type	Description
actionid	ID	Идентификатор действия. Поведение свойства: - *только для чтения*
esc_period	string	- *обязательно* для операций обновления Продолжительность шага операции по умолчанию. Должна составлять не менее 60 секунд. Допускается указать количество секунд (например, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h) или пользовательский макрос. Поведение свойства: - *поддерживается*, если `eventsource` имеет значение ”event created by a trigger”, ”internal event” или ”event created on service status update”

Property	Type	Description
eventsource	integer	Тип событий, которые будет обрабатывать действие. Список поддерживаемых типов событий приведен в описании свойства `source` объекта события. Поведение свойства: - *константа* - *обязательно* для операций создания
name	string	Имя действия. Поведение свойства: - *обязательно* для операций создания
status	integer	Определяет, включено или отключено действие. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.
pause_symptoms	integer	Определяет, приостанавливать ли эскалацию, если событие является событием симптома. Возможные значения: 0 - не приостанавливать эскалацию для проблем-симптомов; 1 - *(по умолчанию)* приостанавливать эскалацию для проблем-симптомов. Поведение свойства: - *поддерживается*, если `eventsource` имеет значение ”event created by a trigger”
pause_suppressed	integer	Определяет, приостанавливать ли эскалацию во время периодов обслуживания. Возможные значения: 0 - не приостанавливать эскалацию; 1 - *(по умолчанию)* приостанавливать эскалацию. Поведение свойства: - *поддерживается*, если `eventsource` имеет значение ”event created by a trigger”
notify_if_canceled	integer	Определяет, отправлять ли уведомление при отмене эскалации. Возможные значения: 0 - не отправлять уведомление при отмене эскалации; 1 - *(по умолчанию)* отправлять уведомление при отмене эскалации. Поведение свойства: - *поддерживается*, если `eventsource` имеет значение ”event created by a trigger”

Операция действия

Объект операции действия определяет операцию, которая будет выполнена при выполнении действия. Он имеет следующие свойства.

Property	Type	Description
operationtype	integer	Тип операции. Возможные значения: 0 - отправка сообщения; 1 - глобальный скрипт; 2 - добавление узла сети; 3 - удаление узла сети; 4 - добавление в группу узлов сети; 5 - удаление из группы узлов сети; 6 - связывание с шаблоном; 7 - отвязывание от шаблона; 8 - включение узла сети; 9 - отключение узла сети; 10 - установка режима инвентаризации узла сети; 13 - добавление тегов узла сети; 14 - удаление тегов узла сети. Возможные значения, если `eventsource` объекта Action имеет значение ”event created by a trigger” или ”event created on service status update”: 0 - ”send message”; 1 - ”global script”. Возможные значения, если `eventsource` объекта Action имеет значение ”internal event”: 0 - ”send message”. Поведение свойства: - *обязательное*
esc_period	string	Продолжительность шага эскалации в секундах. Должна превышать 60 секунд. Принимает количество секунд (например, 60, 300), значение с суффиксом времени (например, `30s`, `1m`, `2h`) или пользовательский макрос. Если задано значение 0 или 0s, будет использоваться период эскалации действия по умолчанию. По умолчанию: 0s. Поведение свойства: - *поддерживается*, если `eventsource` объекта Action имеет значение ”event created by a trigger”, ”internal event” или ”event created on service status update”
esc_step_from	integer	Шаг, с которого начинается эскалация. По умолчанию: 1. Поведение свойства: - *поддерживается*, если `eventsource` объекта Action имеет значение ”event created by a trigger”, ”internal event” или ”event created on service status update”
esc_step_to	integer	Шаг, на котором заканчивается эскалация. По умолчанию: 1. Поведение свойства: - *поддерживается*, если `eventsource` объекта Action имеет значение ”event created by a trigger”, ”internal event” или ”event created on service status update”

Property	Type	Description
evaltype	integer	Метод вычисления условий операции. Возможные значения: 0 - *(по умолчанию)* And/Or; 1 - And; 2 - Or.
opcommand	object	Глобальный скрипт для выполнения. Для глобального скрипта должно быть определено свойство `scriptid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "global script"
opcommand_grp	array	Группы узлов сети, в которых будут выполняться глобальные скрипты. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "global script" и `opcommand_hst` не задано
opcommand_hst	array	Узлы сети, в которых будут выполняться глобальные скрипты. Для узлов сети должно быть определено свойство `hostid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "global script" и `opcommand_grp` не задано
opconditions	array	Условия операции, используемые для действий триггеров. Объект условия операции подробно описан ниже.
opgroup	array	Группы узлов сети, в которые будут добавлены узлы сети. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "add to host group" или "remove from host group"
opmessage	object	Объект, содержащий данные о сообщении, отправляемом операцией. Объект сообщения операции подробно описан ниже. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "send message"
opmessage_grp	array	Группы пользователей, которым будут отправлены сообщения. Для групп пользователей должно быть определено свойство `usrgrpid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "send message" и `opmessage_usr` не задано
opmessage_usr	array	Пользователи, которым будут отправлены сообщения. Для пользователей должно быть определено свойство `userid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение "send message" и `opmessage_grp` не задано

Property	Type	Description
optemplate	array	Шаблоны, которые будут связаны с узлами сети. Для шаблонов должно быть определено свойство `templateid`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение ”link template” или ”unlink template”
opinventory	object	Режим инвентаризации, который будет установлен для узла сети. Для инвентаризации должно быть определено свойство `inventory_mode`. Поведение свойства: - *обязательное*, если для `operationtype` задано значение ”set host inventory mode”
optag	array	Теги узла сети для добавления или удаления. Для тегов должно быть определено свойство `tag`. Свойство `value` является необязательным. Поведение свойства: - *поддерживается*, если для `operationtype` задано значение ”add host tags” или ”remove host tags”.

Сообщение операции действия

Объект сообщения операции содержит данные о сообщении, которое будет отправлено операцией. Он имеет следующие свойства.

Property	Type	Description
default_msg	integer	Использовать ли текст и тему сообщения действия по умолчанию. Возможные значения: 0 - использовать данные из операции; 1 - *(по умолчанию)* использовать данные из типа медиа.
mediatypeid	ID	ID типа медиа, который будет использоваться для отправки сообщения. Поведение свойства: - *поддерживается*, если `operationtype` объекта Action operation object, Action recovery operation object, или Action update operation object установлено в ”send message”, или если `operationtype` объекта Action update operation object установлено в ”notify all involved”
message	string	Текст сообщения операции. Поведение свойства: - *поддерживается*, если `default_msg` установлено в ”use the data from the operation”
subject	string	Тема сообщения операции. Поведение свойства: - *поддерживается*, если `default_msg` установлено в ”use the data from the operation”

Условие операции действия

Объект условия операции действия определяет условие, которое должно быть выполнено для выполнения текущей операции. Он имеет следующие свойства.

Property	Type	Description
conditiontype	integer	Тип условия. Возможные значения: 14 - событие подтверждено. Поведение свойства: - *required*
value	string	Значение для сравнения. Поведение свойства: - *required*
operator	integer	Оператор условия. Возможные значения: 0 - *(default)* =

Для каждого типа условия операции поддерживаются следующие операторы и значения.

Condition	Condition name	Supported operators	Expected value
14	Событие подтверждено	=	Подтверждено ли событие. Возможные значения: 0 - не подтверждено; 1 - подтверждено.

Операция восстановления действия

Объект операции восстановления действия определяет операцию, которая будет выполнена, когда проблема будет устранена. Операции восстановления возможны **только** для действий триггера, внутренних действий и действий сервиса. Он имеет следующие свойства.

Property	Type	Description
operationtype	integer	Тип операции. Возможные значения, если `eventsource` объекта Action object установлен в ”event created by a trigger” или ”event created on service status update”: 0 - отправить сообщение; 1 - глобальный скрипт; 11 - уведомить всех участников. Возможные значения, если `eventsource` объекта Action object установлен в ”internal event”: 0 - отправить сообщение; 11 - уведомить всех участников. Поведение свойства: - *required*
opcommand	object	Глобальный скрипт для выполнения. Глобальный скрипт должен иметь определенное свойство `scriptid`. Поведение свойства: - *required* если `operationtype` установлен в ”global script”

Property	Type	Description
opcommand_grp	array	Группы узлов сети, на которых запускать глобальные скрипты. Группы узлов сети должны иметь определенное свойство `groupid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлен в "event created by a trigger", и `operationtype` установлен в "global script", и `opcommand_hst` не задан
opcommand_hst	array	Узел сети, на котором запускать глобальные скрипты. Узлы сети должны иметь определенное свойство `hostid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлен в "event created by a trigger", и `operationtype` установлен в "global script", и `opcommand_grp` не задан
opmessage	object	Объект, содержащий данные о сообщении, отправляемом операцией восстановления. Объект сообщения операции подробно описан выше. Поведение свойства: - *required* если `operationtype` установлен в "send message"
opmessage_grp	array	Группы пользователей, которым отправлять сообщения. Группы пользователей должны иметь определенное свойство `usrgrpid`. Поведение свойства: - *required* если `operationtype` установлен в "send message" и `opmessage_usr` не задан
opmessage_usr	array	Пользователи, которым отправлять сообщения. Пользователи должны иметь определенное свойство `userid`. Поведение свойства: - *required* если `operationtype` установлен в "send message" и `opmessage_grp` не задан

Операция обновления действия

Объект операции обновления действия определяет операцию, которая будет выполнена при обновлении проблемы (при добавлении комментария, подтверждении, изменении серьезности или ручном закрытии). Операции обновления возможны **только** для действий триггера и сервиса. Он имеет следующие свойства.

Property	Type	Description
operationtype	integer	Тип операции. Возможные значения: 0 - отправить сообщение; 1 - глобальный скрипт; 12 - уведомить всех участников. Поведение свойства: - *required*

Property	Type	Description
opcommand	object	Глобальный скрипт для выполнения. У глобального скрипта должно быть определено свойство `scriptid`. Поведение свойства: - *required* если `operationtype` установлено в ”global script”
opcommand_grp	array	Группы узлов сети, на которых запускать глобальные скрипты. У групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлено в ”event created by a trigger”, и `operationtype` установлено в ”global script”, и `opcommand_hst` не задано
opcommand_hst	array	Узел сети, на котором запускать глобальные скрипты. У узлов сети должно быть определено свойство `hostid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлено в ”event created by a trigger”, и `operationtype` установлено в ”global script”, и `opcommand_grp` не задано
opmessage	object	Объект, содержащий данные о сообщении, отправляемом операцией обновления. Объект сообщения операции подробно описан выше.
opmessage_grp	array	Группы пользователей, которым отправлять сообщения. У групп пользователей должно быть определено свойство `usrgrpid`. Поведение свойства: - *required* если `operationtype` установлено в ”send message” и `opmessage_usr` не задано
opmessage_usr	array	Пользователи, которым отправлять сообщения. У пользователей должно быть определено свойство `userid`. Поведение свойства: - *required* если `operationtype` установлено в ”send message” и `opmessage_grp` не задано

Фильтр действия

Объект фильтра действия определяет набор условий, которые должны быть выполнены для выполнения настроенных операций действия. Он имеет следующие свойства.

Property	Type	Description
conditions	array	Набор условий фильтра, используемых для фильтрации результатов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *required*

Property	Type	Description
evaltype	integer	Метод вычисления условий фильтра. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Поведение свойства: - *required*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для вычисления условий фильтра. Выражение содержит идентификаторы, которые ссылаются на конкретные условия фильтра по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *read-only*
formula	string	Определяемое пользователем выражение, которое будет использоваться для вычисления условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются на конкретные условия фильтра по их `formulaid`. Идентификаторы, используемые в выражении, должны точно соответствовать тем, что определены в условиях фильтра: ни одно условие не может остаться неиспользованным или быть опущенным. Поведение свойства: - *required* если `evaltype` установлено в ”custom expression”

Условие фильтра действия

Объект условия фильтра действия определяет конкретное условие, которое необходимо проверить перед выполнением операций действия.

Свойство	Тип	Описание
conditiontype	integer	Тип условия. Возможные значения, если `eventsource` объекта Action установлен в ”event created by a trigger”: 0 - группа узлов сети; 1 - узел сети; 2 - триггер; 3 - имя события; 4 - важность триггера; 6 - период времени; 13 - шаблон узла сети; 16 - проблема подавлена; 25 - тег события; 26 - значение тега события. Возможные значения, если `eventsource` объекта Action установлен в ”event created by a discovery rule”: 7 - IP-адрес узла сети; 8 - тип обнаруженной службы; 9 - порт обнаруженной службы; 10 - статус обнаружения; 11 - длительность работы или простоя; 12 - полученное значение; 18 - правило обнаружения; 19 - проверка обнаружения; 20 - прокси; 21 - объект обнаружения; 29 - группа прокси. Возможные значения, если `eventsource` объекта Action установлен в ”event created by active agent autoregistration”: 20 - прокси; 22 - имя узла сети; 24 - метаданные узла сети; 29 - группа прокси. Возможные значения, если `eventsource` объекта Action установлен в ”internal event”: 0 - группа узлов сети; 1 - узел сети; 13 - шаблон узла сети; 23 - тип события; 25 - тег события; 26 - значение тега события. Возможные значения, если `eventsource` объекта Action установлен в ”event created on service status update”: 25 - тег события; 26 - значение тега события; 27 - служба; 28 - имя службы. Поведение свойства: - *обязательное*
value	string	Значение для сравнения. Поведение свойства: - *обязательное*

Свойство	Тип	Описание
value2	string	Дополнительное значение для сравнения. Поведение свойства: - *обязательное*, если `eventsource` объекта Action установлен в ”event created by a trigger”, `conditiontype` установлен в любое возможное значение для действий триггера, а тип условия (см. ниже) равен ”26” - *обязательное*, если `eventsource` объекта Action установлен в ”internal event”, `conditiontype` установлен в любое возможное значение для внутренних действий, а тип условия (см. ниже) равен ”26” - *обязательное*, если `eventsource` объекта Action установлен в ”event created on service status update”, `conditiontype` установлен в любое возможное значение для действий службы, а тип условия (см. ниже) равен ”26”
formulaid	string	Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. Идентификатор должен быть задан пользователем при изменении условий фильтра, но при последующем запросе условий будет сгенерирован заново.
operator	integer	Оператор условия. Возможные значения: 0 - *(по умолчанию)* равно; 1 - не равно; 2 - содержит; 3 - не содержит; 4 - входит в список; 5 - больше или равно; 6 - меньше или равно; 7 - не входит в список; 8 - соответствует; 9 - не соответствует; 10 - Да; 11 - Нет.

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, см. примеры на страницах методов action.get и action.create.

Ниже перечислены операторы и значения, поддерживаемые для каждого типа условия.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
0	Группа узлов сети	равно, не равно	Идентификатор группы узлов сети.
1	Узел сети	равно, не равно	Идентификатор узла сети.
2	Триггер	равно, не равно	Идентификатор триггера.
3	Имя события	содержит, не содержит	Имя события.
4	Важность триггера	равно, не равно, больше или равно, меньше или равно	Важность триггера. Список поддерживаемых значений важности триггера см. в описании `свойства severity триггера`.
5	Значение триггера	равно	Значение триггера. Список поддерживаемых значений триггера см. в описании `свойства value триггера`.
6	Период времени	входит в список, не входит в список	Время возникновения события в виде периода времени.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
7	IP-адрес узла сети	равно, не равно	Один или несколько диапазонов IP-адресов для проверки, разделенные запятыми. Дополнительную информацию о поддерживаемых форматах диапазонов IP-адресов см. в разделе настройки обнаружения сети.
8	Тип обнаруженной службы	равно, не равно	Тип обнаруженной службы. Тип службы соответствует типу проверки обнаружения, использованной для обнаружения службы. Список поддерживаемых типов см. в описании свойства `type` проверки обнаружения.
9	Порт обнаруженной службы	равно, не равно	Один или несколько диапазонов портов, разделенных запятыми.
10	Статус обнаружения	равно	Статус обнаруженного объекта. Возможные значения: 0 - узел сети или служба доступны; 1 - узел сети или служба недоступны; 2 - узел сети или служба обнаружены; 3 - узел сети или служба потеряны.
11	Длительность работы или простоя	больше или равно, меньше или равно	Время в секундах, в течение которого обнаруженный объект находится в текущем состоянии.
12	Полученные значения	равно, не равно, больше или равно, меньше или равно, содержит, не содержит	Значение, возвращенное при выполнении проверки обнаружения с помощью агента Zabbix, SNMPv1, SNMPv2 или SNMPv3.
13	Шаблон узла сети	равно, не равно	Идентификатор связанного шаблона.
16	Проблема подавлена	Да, Нет	Значение не требуется: оператор ”Да” означает, что проблема должна быть подавлена, а ”Нет” - что проблема не должна быть подавлена.
18	Правило обнаружения	равно, не равно	Идентификатор правила обнаружения.
19	Проверка обнаружения	равно, не равно	Идентификатор проверки обнаружения.
20	Прокси	равно, не равно	Идентификатор прокси.
21	Объект обнаружения	равно	Тип объекта, вызвавшего событие обнаружения. Возможные значения: 1 - обнаруженный узел сети; 2 - обнаруженная служба.
22	Имя узла сети	содержит, не содержит, соответствует, не соответствует	Имя узла сети. Использование регулярного выражения поддерживается для операторов *соответствует* и *не соответствует* в условиях автоматической регистрации.
23	Тип события	равно	Конкретное внутреннее событие. Возможные значения: 0 - элемент данных в состоянии ”не поддерживается”; 1 - элемент данных в состоянии ”норма”; 2 - правило LLD в состоянии ”не поддерживается”; 3 - правило LLD в состоянии ”норма”; 4 - триггер в состоянии ”неизвестно”; 5 - триггер в состоянии ”норма”.
24	Метаданные узла сети	содержит, не содержит, соответствует, не соответствует	Метаданные автоматически зарегистрированного узла сети. Использование регулярного выражения поддерживается для операторов *соответствует* и *не соответствует*.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
25	Тег	равно, не равно, содержит, не содержит	Тег события.
26	Значение тега	равно, не равно, содержит, не содержит	Значение тега события.
27	Служба	равно, не равно	Идентификатор службы.
28	Имя службы	равно, не равно	Имя службы.
29	Группа прокси	равно, не равно	Идентификатор группы прокси.

action.create

Описание

`object action.create(object/array actions)`

Этот метод позволяет создавать новые действия.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object/array)` Действия для создания.

В дополнение к стандартным свойствам действия, метод принимает следующие параметры.

Parameter	Type	Description
filter	object	Объект фильтра действия для действия.
operations	array	Операции действия для создания для действия.
recovery_operations	array	Операции восстановления действия для создания для действия.
update_operations	array	Операции обновления действия для создания для действия.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных действий под свойством `actionids`. Порядок возвращаемых ID соответствует порядку переданных действий.

Примеры

Создание действия триггера

Создать действие триггера, которое начнется, как только триггер (со словом «память» в названии) с хоста «10084» перейдет в состояние ПРОБЛЕМА. Действие будет иметь 4 настроенные операции. Первая и немедленная операция отправит сообщение всем пользователям в группе пользователей «7» через тип носителя «1». Если событие не будет устранено в течение 30 минут, вторая операция запустит script ”5” (скрипт с областью действия ”Операция действия”) на всех хостах группы «2». Если событие устранено, операция восстановления уведомит всех пользователей, получивших сообщения о проблеме. Если событие обновлено, операция подтверждения/обновления уведомит (с настраиваемой темой и сообщением) всех пользователей, получивших какие-либо сообщения о проблеме.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "30m",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084"
                },
                {
                    "conditiontype": 3,
                    "operator": 2,
                    "value": "memory"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            },
            {
                "operationtype": 1,
                "esc_step_from": 2,
                "esc_step_to": 2,
                "opconditions": [
                    {
                        "conditiontype": 14,
                        "operator": 0,
                        "value": "0"
                    }
                ],
                "opcommand_grp": [
                    {
                        "groupid": "2"
                    }
                ],
                "opcommand": {
                    "scriptid": "5"
                }
            }
        ],
        "recovery_operations": [
            {
                "operationtype": "11",
                "opmessage": {
```

```
                    "default_msg": 1
                }
            }
        ],
        "update_operations": [
            {
                "operationtype": "12",
                "opmessage": {
                    "default_msg": 0,
                    "message": "Custom update operation message body",
                    "subject": "Custom update operation message subject"
                }
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Создание действия обнаружения

Создать действие обнаружения, которое свяжет шаблон «10001» с обнаруженными хостами.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Discovery action",
        "eventsource": 1,
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 21,
                    "operator": 0,
                    "value": "1"
                },
                {
                    "conditiontype": 10,
                    "operator": 0,
                    "value": "2"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 6,
                "optemplate": [
                    {
                        "templateid": "10001"
                    }
                ]
```

```
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Использование пользовательского фильтра выражений

Создать действие триггера, использующее пользовательское выражение "A и (B или C)" для оценки условий действия. Как только триггер с уровнем серьезности выше или равным "Предупреждение" с хоста "10084" или хоста "10106" переходит в состояние ПРОБЛЕМА, действие отправит сообщение всем пользователям в группе пользователей "7" через тип носителя "1". Идентификаторы формул "A", "B" и "C" были выбраны произвольно.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "15m",
        "filter": {
            "evaltype": 3,
            "formula": "A and (B or C)",
            "conditions": [
                {
                    "conditiontype": 4,
                    "operator": 5,
                    "value": "2",
                    "formulaid": "A"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084",
                    "formulaid": "B"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10106",
                    "formulaid": "C"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
```

```
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Создание правила автоматической регистрации агента

Создать действие авторегистрации, которое добавляет хост в группу хостов «2», если имя хоста содержит «SRV» или метаданные содержат ”AlmaLinux”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "id": 1
```

```
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

Создание правила автрегистрции агента с тегами узла сети

Создайте действие автрегистрции, которое добавляет узел сети в группу узлов сети "2" и добавляет два тега узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers with tags",
        "eventsource": "2",
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            },
            {
                "operationtype": "13",
                "optag": [
                    {
                        "tag": "location",
                        "value": "office"
                    },
                    {
                        "tag": "city",
                        "value": "Riga"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

См. также

- Фильтр действия
- Операция действия
- Скрипт

Источник

CAction::create() в *ui/include/classes/api/services/CAction.php*.

action.delete

Описание

`object action.delete(array actionIds)`

Этот метод позволяет удалять действия.

> **Note:**
> Этот метод доступен только пользователям с типом *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.
> Пользователь не сможет удалить действие, содержащее недоступный прокси или группу прокси в условиях фильтра. Дополнительную информацию см. в разделе Настройка групп пользователей (*Список доступа к прокси*).

Параметры

`(array)` ID удаляемых действий.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID удаленных действий под свойством `actionids`.

Примеры

Удаление нескольких действий

Удалить два действия

Request:

```
{
 · "jsonrpc": "2.0",
 · "method": "action.delete",
 · "params": [
 · "17",
 · "18"
 · ],
 · "auth": "3a57200802b24cda67c4e4010b50c065",
 · "id": 1
}
```

Response:

```
{
 · "jsonrpc": "2.0",
 · "result": {
 · "action ids": [
 · "17",
 · "18"
 · ]
 · },
 · "id": 1
}
```

Источник

CAction::delete() в *ui/include/classes/api/services/CAction.php*.

action.get

Описание

`integer/array action.get(object parameters)`

Метод позволяет получать действия в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
actionids	ID/array	Возвращать только действия с указанными ID.
groupids	ID/array	Возвращать только действия, которые используют указанные группы узлов сети в условиях действия.
hostids	ID/array	Возвращать только действия, которые используют указанные узлы сети в условиях действия.
triggerids	ID/array	Возвращать только действия, которые используют указанные триггеры в условиях действия.
mediatypeids	ID/array	Возвращать только действия, которые используют указанные типы медиа для отправки сообщений.
usrgrpids	ID/array	Возвращать только действия, которые настроены на отправку сообщений указанным группам пользователей.
userids	ID/array	Возвращать только действия, которые настроены на отправку сообщений указанным пользователям.
scriptids	ID/array	Возвращать только действия, которые настроены на запуск указанных скриптов.
selectFilter	query	Возвращать свойство `filter` с фильтром условий действия.
selectOperations	query	Возвращать свойство `operations` с операциями действия.
selectRecoveryOperations	query	Возвращать свойство `recovery_operations` с операциями восстановления действия.
selectUpdateOperations	query	Возвращать свойство `update_operations` с операциями обновления действия.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `actionid`, `name`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(  /  )` Возвращает:

- массив объектов

или - количество полученных объектов , если был использован параметр `countOutput`.

Примеры

Получение действий триггеров

Получить все настроенные действия триггеров вместе с условиями действия и операциями.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Report problems to Zabbix administrators",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
```

```
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nTrigger state: {TRIGGER.STATUS}\r\nTrigger
                        "mediatypeid": "0"
                    }
                }
            ],
            "update_operations": [
                {
                    "operationid": "31",
                    "operationtype": "12",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "1",
                        "subject": "",
                        "message": "",
                        "mediatypeid": "0"
                    }
                },
                {
                    "operationid": "32",
                    "operationtype": "0",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "Updated: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                        "mediatypeid": "1"
                    },
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ],
                    "opmessage_usr": []
                },
                {
                    "operationid": "33",
                    "operationtype": "1",
                    "evaltype": "0",
                    "opcommand": {
                        "scriptid": "3"
                    },
                    "opcommand_hst": [
                        {
                            "hostid": "10084"
                        }
                    ],
                    "opcommand_grp": []
                }
            ]
        }
    ],
    "id": 1
}
```


Получение действий обнаружения

Вернуть все настроенные действия обнаружения вместе с условиями и операциями этих действий. Фильтр использует

тип вычисления "и", поэтому свойство `formula` является пустым, а свойство`eval_formula` генерируется автоматически.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
    "output": "extend",
    "selectOperations": "extend",
    "selectFilter": "extend",
    "filter": {
    "eventsource": 1
    }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
    {
    "actionid": "2",
    "name": "Auto discovery. Linux servers.",
    "eventsource": "1",
    "status": "1",
    "esc_period": "0s",
    "pause_suppressed": "1",
    "filter": {
    "evaltype": "0",
    "formula": "",
    "conditions": [
    {
    "conditiontype": "10",
    "operator": "0",
    "value": "0",
    "value2": "",
    "formulaid": "B"
    },
    {
    "conditiontype": "8",
    "operator": "0",
    "value": "9",
    "value2": "",
    "formulaid": "C"
    },
    {
    "conditiontype": "12",
    "operator": "2",
    "value": "Linux",
    "value2": "",
    "formulaid": "A"
    }
    ],
    "eval_formula": "A and B and C"
    },
    "operations": [
    {
    "operationid": "1",
    "actionid": "2",
    "operationtype": "6",
    "esc_period": "0s",
    "esc_step_from": "1",
```

```
    "esc_step_to": "1",
    "evaltype": "0",
    "opconditions": [],
    "optemplate": [
    {
    "templateid": "10001"
    }
    ]
    },
    {
    "operationid": "2",
    "actionid": "2",
    "operationtype": "4",
    "esc_period": "0s",
    "esc_step_from": "1",
    "esc_step_to": "1",
    "evaltype": "0",
    "opconditions": [],
    "opgroup": [
    {
    "groupid": "2"
    }
    ]
    }
    ]
    }
    ],
    "id": 1
}
```


См. также

- Фильтр действия
- Операция действия

Источник

CAction::get() в *ui/include/classes/api/services/CAction.php*.

action.update

Описание

`object action.update(object/array actions)`

Этот метод позволяет обновлять существующие действия.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства действия, которые необходимо обновить.

Свойство `actionid` должно быть определено для каждого действия, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам действия, метод принимает следующие параметры.

Parameter	Type	Description
filter	object	Объект фильтра действия для замены текущего фильтра.
operations	array	Операции действия для замены существующих операций.

Parameter	Type	Description
recovery_operations	array	Операции восстановления действия для замены существующих операций восстановления. Поведение параметра: - *supported*, если `eventsource` объекта действия установлен в ”event created by a trigger”, ”internal event” или ”event created on service status update”
update_operations	array	Операции обновления действия для замены существующих операций обновления. Поведение параметра: - *supported*, если `eventsource` объекта действия установлен в ”event created by a trigger” или ”event created on service status update”

Возвращаемые значения

`(object)` Возвращает объект, содержащий ID обновленных действий под свойством `actionids`.

Примеры

Деактивация действия

Деактивировать действие, то есть, выставить его статус в ”1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Фильтр действия
- Операция действия

Источник

CAction::update() в *ui/include/classes/api/services/CAction.php*.

Alert

Этот класс предназначен для работы с оповещениями.

Ссылки на объекты:

- ·Alert

Доступные методы:

- alert.get - получение предупреждений

Объект оповещение

Следующие объекты напрямую связаны с методом API `alert`.

Оповещение

Note:
Оповещения создаются сервером Zabbix и не могут быть изменены через API.

Объект alert содержит информацию о том, были ли успешно выполнены определенные операции действия. Он имеет следующие свойства.

Property	Type	Description
alertid	ID	ID оповещения.
actionid	ID	ID действия, которое сгенерировало оповещение.
alerttype	integer	Тип оповещения. Возможные значения: 0 - сообщение; 1 - удаленная команда.
clock	timestamp	Время, когда было сгенерировано оповещение.
error	string	Текст ошибки, если возникли проблемы при отправке сообщения или выполнении команды.
esc_step	integer	Шаг эскалации действия, во время которого было сгенерировано оповещение.
eventid	ID	ID события, которое запустило действие.
mediatypeid	ID	ID типа медиа, который был использован для отправки сообщения.
message	text	Текст сообщения. Property behavior: - *supported* if `alerttype` is set to ”message”
retries	integer	Количество попыток, которые Zabbix предпринял для отправки сообщения.
sendto	string	Адрес, имя пользователя или другой идентификатор получателя. Property behavior: - *supported* if `alerttype` is set to ”message”
status	integer	Статус, указывающий, было ли успешно выполнено операцию действия. Возможные значения, если `alerttype` is set to ”message”: 0 - сообщение не отправлено; 1 - сообщение отправлено; 2 - сбой после нескольких повторных попыток; 3 - новое оповещение еще не обработано менеджером оповещений. Возможные значения, если `alerttype` is set to ”remote command”: 0 - команда не выполнена; 1 - команда выполнена; 2 - была предпринята попытка выполнить команду на агенте Zabbix, но он был недоступен.
subject	string	Тема сообщения. Property behavior: - *supported* if `alerttype` is set to ”message”
userid	ID	ID пользователя, которому было отправлено сообщение.
p_eventid	ID	ID проблемного события, которое сгенерировало оповещение.

Property	Type	Description
acknowledgeid	ID	ID подтверждения, которое сгенерировало оповещение.

alert.get

Описание

`integer/array alert.get(object parameters)`

Метод позволяет получать оповещения в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
alertids	ID/array	Возвращать только оповещения с указанными ID.
actionids	ID/array	Возвращать только оповещения, сгенерированные указанными действиями.
eventids	ID/array	Возвращать только оповещения, сгенерированные указанными событиями.
groupids	ID/array	Возвращать только оповещения, сгенерированные объектами из указанных групп узлов сети.
hostids	ID/array	Возвращать только оповещения, сгенерированные объектами из указанных узлов сети.
mediatypeids	ID/array	Возвращать только оповещения сообщений, использовавшие указанные типы медиа.
objectids	ID/array	Возвращать только оповещения, сгенерированные указанными объектами
userids	ID/array	Возвращать только оповещения сообщений, отправленные указанным пользователям.
eventobject	integer	Возвращать только оповещения, сгенерированные событиями, связанными с объектами указанного типа. См. event `object` для списка поддерживаемых типов объектов. По умолчанию: 0 - триггер.
eventsource	integer	Возвращать только оповещения, сгенерированные событиями указанного типа. См. event `source` для списка поддерживаемых типов событий. По умолчанию: 0 - события триггера.
time_from	timestamp	Возвращать только оповещения, сгенерированные после указанного времени.
time_till	timestamp	Возвращать только оповещения, сгенерированные до указанного времени.
selectHosts	query	Возвращать свойство `hosts` с данными узлов сети, которые вызвали операцию действия.
selectMediatypes	query	Возвращать свойство `mediatypes` с массивом типов медиа, использованных для оповещения сообщения. См. `mediatype.get` для ограничений в зависимости от типа пользователя.

Parameter	Type	Description
selectUsers	query	Возвращать свойство `users` с массивом пользователей, которым было адресовано сообщение. См. `user.get` для ограничений в зависимости от типа пользователя.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение оповещений по ID действия

Получить все оповещения, сгенерированные действием ”3”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
```

```
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```


См. также

- Узел сети
- Тип медиа
- Пользователь

Источник

CAlert::get() в *ui/include/classes/api/services/CAlert.php*.

API info

Этот класс предназначен для получения метаинформации об API.

Доступные методы:

- apiinfo.version - получение версии Zabbix API

apiinfo.version

Описание

`string apiinfo.version(array)`

Этот метод позволяет получить версию Zabbix API.

> **Attention:**
> Этот метод доступен только неаутентифицированным пользователям.

Параметры

`(array)` Метод принимает пустой массив.

Return values

`(string)` Возвращает версию Zabbix API.

> **Note:**
> Начиная с Zabbix 2.0.4 версия API соответствует версии Zabbix.

Примеры

Получение версии API

Получите версию Zabbix API.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```


Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Источник

CAPIInfo::version() в *ui/include/classes/api/services/CAPIInfo.php*.

APM span

Этот класс предназначен для работы с интервалами, собранными для мониторинга производительности приложений (APM).

Ссылки на объекты:

- APM span

Доступные методы:

- apmspan.get - получение интервалов

Объект интервала APM

Следующие объекты напрямую связаны с API `apmspan`.

Note:
Каждое свойство соответствует столбцу в таблице `otel_traces` источника данных телеметрии. Список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии.

Спан APM

Объект спана APM представляет спан (одну операцию внутри трейса, например HTTP-запрос, запрос к базе данных или вызов функции).

Спаны образуют иерархию, начиная с корневого спана, который представляет первую операцию запроса. Каждый другой спан является дочерним спаном того спана, во время которого была запущена его операция.

Объект спана APM имеет следующие свойства:

Свойство	Тип	Описание
timestamp	integer	Время в формате Unix (в наносекундах), когда был запущен спан. Поведение свойства: - *только для чтения*
traceid	string	ID трейса, к которому относится спан. Поведение свойства: - *только для чтения*
spanid	string	ID спана. Поведение свойства: - *только для чтения*
parent_spanid	string	ID родительского спана. Для корневого спана значение пустое. Поведение свойства: - *только для чтения*
trace_state	string	Дополнительные данные, если они есть, которые инструменты трассировки добавляют к трейсу (см. контекст трейса W3C). Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
span_name	string	Имя операции, которую представляет спан (например, `mysqli_query`). Поведение свойства: - *только для чтения*
span_kind	string	Роль спана. Возможные значения: `Unspecified` - тип спана не указан; `Internal` - операция, выполненная внутри приложения; `Server` - обработка входящего запроса; `Client` - исходящий запрос; `Producer` - создание асинхронной операции (например, отправка сообщения); `Consumer` - обработка асинхронной операции. Поведение свойства: - *только для чтения*
service_name	string	Имя сервиса, создавшего спан (например, `zabbix-frontend`), взятое из атрибута ресурса OpenTelemetry `service.name`. Поведение свойства: - *только для чтения*
resource_attributes	object	Атрибуты ресурса спана в формате «ключ-значение» (например, `host.name: eabf8c2a5ddc`). Поведение свойства: - *только для чтения*
scope_name	string	Имя области инструментирования, создавшей спан. Поведение свойства: - *только для чтения*
scope_version	string	Версия области инструментирования. Поведение свойства: - *только для чтения*
span_attributes	object	Атрибуты спана в формате «ключ-значение» (например, `db.system: mysql`). Поведение свойства: - *только для чтения*
duration	integer	Продолжительность спана в наносекундах. Поведение свойства: - *только для чтения*
status_code	string	Статус операции, которую представляет спан. Возможные значения: `Unset` - спану не был явно назначен статус; `Ok` - операция успешно завершена; `Error` - операция завершена с ошибкой. Поведение свойства: - *только для чтения*
status_message	string	Дополнительная информация, если она есть, о статусе спана. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
events	array	Массив объектов событий спана, то есть значимых моментов времени во время операции (например, исключений). Поведение свойства: - *только для чтения*
links	array	Массив объектов связей спана, указывающих на связанные спаны в том же или другом трейсе. Поведение свойства: - *только для чтения*

apmspan.get

Описание

`integer/array apmspan.get(object parameters)`

Метод позволяет получать спаны в соответствии с указанными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
time_from	timestamp	Возвращать только интервалы, начавшиеся в указанное время или позже. Поведение параметра: - *обязательный*
time_till	timestamp	Возвращать только интервалы, начавшиеся до указанного времени. Поведение параметра: - *обязательный*
traceids	string/array	Возвращать только интервалы, относящиеся к указанным трассировкам.
spanids	string/array	Возвращать только интервалы с указанными идентификаторами.
parent_spanids	string/array	Возвращать только интервалы с указанными родительскими интервалами.
resource_attributes	object/array	Возвращать только интервалы с указанными атрибутами ресурса. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все интервалы. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.

Параметр	Тип	Описание
resource_attributes_evaltype	integer	Метод оценки условий в `resource_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
span_attributes	object/array	Возвращать только интервалы с указанными атрибутами интервала. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все интервалы. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
span_attributes_evaltype	integer	Метод оценки условий в `span_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
min_duration	integer	Возвращать только интервалы, длительность которых больше или равна указанному значению в наносекундах.
max_duration	integer	Возвращать только интервалы, длительность которых меньше указанного значения в наносекундах.
filter	object	Возвращать только результаты, которые точно соответствуют указанному фильтру. Принимает объект, в котором ключами являются имена свойств, а значениями - отдельное значение или массив значений для сопоставления. Поддерживаемые свойства: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, в котором ключами являются имена свойств, а значениями - строки для поиска. Если дополнительные параметры не указаны, выполняется поиск `LIKE "%…%"`. Поддерживаемые свойства: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Количество записей, которые необходимо пропустить перед возвратом результата.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Параметр	Тип	Описание
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение иерархии спанов трассировки

Получите первые четыре спана трассировки "dee625f33cb0994e86c5cd4640e3b09c", отсортированные по времени начала. Свойство `parent_spanid` показывает взаимосвязь спанов: у корневого спана нет родительского спана, а каждый другой спан ссылается на спан, который его запустил.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
```

```
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```


См. также

- Трассировка APM
- Журнал APM
- Метрика APM

Источник

CApmSpan::get() в файле *ui/include/classes/api/services/CApmSpan.php*.

Complex event processing

This class is designed to work with complex event processing.

Object references:

- CEP rule
- CEP rule filter
  - CEP rule filter condition
- CEP rule time window
- CEP rule operation
- CEP rule operation filter
- CEP rule operation filter condition

Available methods:

- ceprule.create - create new CEP rules
- ceprule.delete - delete CEP rules
- ceprule.get - retrieve CEP rules
- ceprule.update - update CEP rules
- ceprule.resettimewindows - reset time windows for CEP rule

CEP rule object

The following objects are directly related to the `ceprule` API.

CEP rule

The CEP rule object has the following properties.

Property	Type	Description
cep_ruleid	ID	ID of the CEP rule. Property behavior: - *read-only* - *required* for update operations
name	string	Name of the CEP rule. Property behavior: - *required* for create operations

Property	Type	Description
window_type	integer	Time window type: 0 - (default) None 1 - Simple (multiple); 2 - Cause and symptom grouping; 3 - Simple (once); 4 - Event pattern match.
stop	integer	Possible values: 0 - (default) Continue with the next CEP rule according to sort order; 1 - Stop after executing this rule.
sortorder	integer	Execution order of the CEP rule.
description	string	Description of the CEP rule.
status	integer	Whether the CEP rule is enabled or disabled. Possible values: 0 - *(default)* enabled; 1 - disabled.
error	string	Error text if the CEP rule processing failed on rule level. Property behavior: - *read-only*

CEP rule filter

The CEP rule filter object defines a set of conditions that must be met for events to be processed by CEP rule. It has the following properties.

Property	Type	Description
conditions	array	Set of filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	User-defined expression to be used for evaluating conditions of filters with a custom expression. The expression must contain IDs that reference specific filter conditions by its `formulaid`. The IDs used in the expression must exactly match the ones defined in the filter conditions: no condition can remain unused or omitted. Property behavior: - *required* if `evaltype` is set to "custom expression"

CEP rule filter condition

The CEP rule filter condition object defines a specific condition that must be checked before processing events by the CEP rule.

Property	Type	Description
type	integer	Type of condition. Possible values: 3 - event name; 4 - severity; 6 - time period; 25 - tag name; 26 - tag value; 36 - host visible name; 37 - host group name. Property behavior: - *required*
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 4 - in; 5 - is greater than or equals; 6 - is less than or equals; 7 - not in. Property behavior: - *required*.
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”event name”
severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
time_period	string	Time period for a specific operation type. Property behavior: - *required* if `type` is set to ”time period”
tag	string	Tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”tag name” or ”tag value”
tag_value	string	Tag value for a specific operation type. Property behavior: - *required* if `type` is set to ”tag value”
event_severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
host_name	string	Host name for a specific operation type. Property behavior: - *required* if `type` is set to ”host visible name”

Property	Type	Description
host_group_name	string	Host group name for a specific operation type. Property behavior: - *required* if `type` is set to ”host group name”
formulaid	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required* if CEP rule filter `evaltype` is set to ”custom expression”

CEP rule time window

The CEP rule time window object defines the time window of the CEP rule. It has the following properties.

Property	Type	Description
duration	string	Duration of the CEP rule window (365 days maximum). Accepts seconds and time units with suffix. Default: `10m`
capacity	integer	Capacity of the CEP rule window. Supports user macros. Default: 0 (unlimited)
group_by_host_group	integer	Group by flag for host group. Possible values: 0 - no (default); 1 - yes.
group_by_host	integer	Group by flag for host. Possible values: 0 - no (default); 1 - yes.
group_by_tags	integer	Group by flag for tag. Possible values: 0 - no (default); 1 - yes.
tags	string	Tag names to use for grouping. Property behavior: - *required* if `group_by_tag` is set to ”1”
event_count_tag	string	Tag name for displaying the count of affected events. Property behavior: - *supported* if CEP rule `window_type` is set to ”2”; not allowed otherwise
script	string	Script to execute. Property behavior: - *supported* if CEP rule `window_type` is set to ”4”; not allowed otherwise

CEP rule operation

The CEP rule operation object defines an operation that will be performed when a CEP rule is executed. It has the following properties.

Property	Type	Description
sortorder	integer	Execution order (ascending) of the operation. Property behavior: - *required*
execute_when	integer	Type of condition for operation. Possible values: 0 - event occurred; 1 - event added; 2 - event evicted; 3 - window closed; 4 - event pattern matched; 5 - event closed. Property behavior: - *required*
scope	integer	Scope of the operation. Possible values: 0 - current event (default); 1 - other events; 2 - all events. Possible values when `execute_when` is ”Window closed” or ”Pattern matched”: 2 - all events. Possible values when `execute_when` is ”Event ocurred”: 0 - current event.
filter	object	Filter object.
scope_filter	object	Scope filter object. scope_filter is editable only if the `scope` field contains ”Other events” or ”All events” value. Otherwise scope_filter accepts only object with default values.

Property	Type	Description
type	integer	CEP operation type. Possible values: 1 - set event name; 2 - close event; 3 - discard event; 4 - set event severity; 5 - increase event severity; 6 - decrease event severity; 7 - suppress event; 8 - unsuppress event; 9 - copy first event as new; 10 - copy last event as new; 11 - add tag; 12 - set tag; 13 - set tag value; 14 - increase tag value; 15 - decrease tag value; 16 - rename tag; 17 - remove tag; 18 - close window. Property behavior: - *required* - *supported*: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17 - supported if ”Event occurred” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event added” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event evicted” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 - supported if ”Window closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event pattern matched” (see `execute_when`)
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event name”
tag	string	Tag name for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
new_tag	string	New tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”Rename tag”
tag_value	string	Tag value for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
severity	integer	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event severity”

Property	Type	Description
suppress_duration	string	Duration of the event suppression. Accepts seconds and time unit with suffix. Property behavior: - *required* if `type` is set to ”Suppress event”

CEP rule operation filter

The CEP rule operation filter object defines a set of conditions that must be met to perform the configured CEP operations. It has the following properties.

Property	Type	Description
conditions	array	Set of operation filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation filter condition

The CEP rule operation filter condition object defines a specific condition that must be checked before running the CEP rule operations.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

CEP rule operation scope filter

The CEP rule operation scope filter object defines a set of conditions that must be met to apply the configured CEP operations to events. It has the following properties.

Property	Type	Description
conditions	array	Set of operation scope filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*

Property	Type	Description
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation scope filter condition

The CEP rule operation scope filter condition object defines a specific condition that must be checked before applying the CEP rule operations to events.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

ceprule.create

Description

`object ceprule.create(object/array CEP rules)`

This method allows you to create new CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rules to create.

Additionally to the standard CEP properties, the method accepts the following parameters.

Parameter	Type	Description
operations	array	CEP rule operations to create for the CEP rule. Parameter behavior: - *required*
filter	object	CEP rule filter object for the CEP rule. Parameter behavior: - *required*

Return values

`(object)` Returns an object containing the IDs of the created CEP rules under the `cep_ruleids` property. The order of the returned IDs matches the order of the passed CEP rules.

Examples

Create a new CEP rule

Create a CEP rule using evaluation method `AND/OR` with one condition and one window. By default, the rule will be enabled.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.create",
    "params": {
      "name": "Group events by cause",
      "stop": "1",
      "sortorder": "0",
      "description": "",
      "status": "0",
      "window_type": "2",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": ""
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "",
        "script": ""
      },
      "operations": []
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "cep_ruleids": [
      "1"
    ]
  },
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::create() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.delete

Description

`object ceprule.delete(array cep_ruleids)`

This method allows you to delete CEP rules.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(array)` IDs of the CEP rules to delete.

Return values

`(object)` Returns an object containing the IDs of the deleted CEP rules under the `cep_ruleids` property.

Example

Delete multiple CEP rules

Delete two CEP rules.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Source

CCepRule::delete() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.get

Description

`integer/array ceprule.get(object parameters)`

The method allows you to retrieve CEP rules according to the given parameters.

Note:
This method is available to users of any type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` Parameters defining the desired output.

The method supports the following parameters.

Parameter	Type	Description
cep_ruleids	ID/array	Return only CEP rules with the given IDs.
selectFilter	query	Return a `filter` property with the CEP rule conditions.
selectOperations	query	Return an `operations` property with the CEP rule operations.
selectWindow	query	Return a `[window]`(/manual/api/reference/ceprule/object#cep-rule-time-window) property with the CEP rule time window.
sortfield	string/array	Sort the result by the given properties. Possible values: `cep_ruleid`, `name`, `stop`, `sortorder`, `status`.
countOutput	boolean	These parameters are described in the reference commentary.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Return values

`(integer/array)` Returns either:

- an array of objects;
- the count of retrieved objects, if the `countOutput` parameter has been used.

Examples

Retrieve CEP rules

Retrieve enabled CEP rules, including conditions, windows, and operations.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.get",
    "params": {
        "output": "extend",
        "filter": {"status": 0},
        "selectFilter": "extend",
        "selectWindow": "extend",
        "selectOperations": "extend"
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "cep_ruleid": "2",
```

```
      "name": "Group events by cause",
      "stop": "0",
      "sortorder": "1",
      "description": "",
      "status": "0",
      "window_type": "2",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": "A"
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "symptoms",
        "script": ""
      },
      "operations": []
    },
    {
      "cep_ruleid": "10",
      "name": "Suppress consecutive events",
      "stop": "1",
      "sortorder": "2",
      "description": "",
      "status": "0",
      "window_type": "3",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
```

```
          "host_group_name": "",
          "time_period": "",
          "formulaid": "A"
        }
      ]
    },
    "window": {
      "duration": "3m",
      "capacity": "0",
      "group_by_host_group": "0",
      "group_by_host": "1",
      "group_by_tags": "1",
      "tags": [
        "cause"
      ],
      "event_count_tag": "",
      "script": ""
    },
    "operations": [
      {
        "sortorder": "1",
        "execute_when": "1",
        "type": "7",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "6m",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
        "filter": {
          "evaltype": "0",
          "eval_formula": "A",
          "formula": "",
          "conditions": [
            {
              "type": "33",
              "operator": "10",
              "tag": "",
              "tag_value": "",
              "formulaid": "A"
            }
          ]
        },
        "scope_filter": {
          "evaltype": "0",
          "eval_formula": "",
          "formula": "",
          "conditions": []
        }
      },
      {
        "sortorder": "2",
        "execute_when": "2",
        "type": "8",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "0",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
```

```
          "filter": {
            "evaltype": "0",
            "eval_formula": "A",
            "formula": "",
            "conditions": [
              {
                "type": "35",
                "operator": "10",
                "tag": "",
                "tag_value": "",
                "formulaid": "A"
              }
            ]
          },
          "scope_filter": {
            "evaltype": "0",
            "eval_formula": "",
            "formula": "",
            "conditions": []
          }
        }
      ]
    }
  ],
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::get() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.resettimewindows

Description

`object ceprule.resettimewindows(object/array cep_ruleid)`

This method allows you to reset time windows for an existing CEP rule.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` CEP rule object must have a `cep_ruleid` property defined.

Return values

`(object)` Returns an object containing the ID of the CEP rule whose time windows have been reset under the `cep_ruleid` property.

Examples

Reset time windows for CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.resettimewindows",
    "params": {
        "cep_ruleid": "1"
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "cep_ruleid": "1"
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::resettimewindows() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.update

Description

`object ceprule.update(object/array CEP rules)`

This method allows you to update existing CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rule properties to be updated.

The `cep_ruleid` property must be defined for each CEP rule, all other properties are optional. Only the passed properties will be updated, all others will remain unchanged.

Additionally to the standard CEP rule properties, the method accepts the following parameters.

Parameter	Type	Description
filter	object	CEP rule filter object to replace the current filter.
window	array	CEP rule window to be updated. Property behavior: - *required* if `window_type` is changed to ”Cause and symptom grouping” and ”group_by_host_group”, ”group_by_host”, ”group_by_tags” are all set to NO; - *required* if `window_type` is changed to ”Event pattern match”; - *supported* if `window_type` is ”Simple (multiple)” or ”Simple (once)”
operations	array	CEP rule operations to replace existing operations. Property behavior: - *required* if `window_type` is changed to ”None” (0), ”Simple (multiple)” (1), ”Simple (once)” (3) or ”Event pattern match” (4).

Return values

`(object)` Returns an object containing the IDs of the updated CEP rules under the `cep_ruleids` property.

Examples

Disable CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "1",
        "status": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

Replace condition

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "3",
        "filter": {
        "evaltype": "0",
        "formula": "",
            "conditions": [
                {
                    "type": "3",
                    "operator": "2",
                    "event_name": "Inaccessible port",
                    "severity": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::update() in *ui/include/classes/api/services/CCepRule.php*.

Connector

Этот класс предназначен для работы с коннекторами.

Ссылки на объекты:

- Connector
- Tag filter

Доступные методы:

- connector.create - создать новые коннекторы
- connector.delete - удалить коннекторы
- connector.get - получить коннекторы
- connector.update - обновить коннекторы

Объект connector

Следующие объекты напрямую связаны с API `connector`.

Коннектор

Объект коннектора имеет следующие свойства.

Property	Type	Description
connectorid	ID	Идентификатор коннектора. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя коннектора. Поведение свойства: - *обязательно* для операций создания
url	string	URL конечной точки, то есть URL получателя. Поддерживаются пользовательские макросы. Поведение свойства: - *обязательно* для операций создания
protocol	integer	Протокол связи. Возможные значения: 0 - *(по умолчанию)* Zabbix Streaming Protocol v1.0.
data_type	integer	Тип данных. Возможные значения: 0 - *(по умолчанию)* Значения элементов данных; 1 - События.

Property	Type	Description
item_value_type	integer	Сумма типов значений элементов данных, которые необходимо отправлять. Возможные значения: 1 - Числовой (с плавающей запятой); 2 - Символьный; 4 - Журнал; 8 - Числовой (без знака); 16 - Текстовый; 32 - Двоичный; 64 - JSON. По умолчанию: 31 - Все типы элементов данных (кроме двоичных и JSON). Поведение свойства: - *поддерживается*, если для `data_type` установлено значение «Значения элементов данных».
max_records	integer	Максимальное количество событий или элементов данных, которые можно отправить в одном сообщении. Возможные значения: 0-2147483647 (максимальное значение 32-разрядного целого числа со знаком). По умолчанию: 0 - Без ограничений.
max_senders	integer	Количество процессов отправки, запускаемых для этого коннектора. Возможные значения: 1-100. По умолчанию: 1.
max_attempts	integer	Количество попыток. Возможные значения: 1-5. По умолчанию: 1.
attempt_interval	string	Интервал между повторными попытками. Принимает значения в секундах. Возможные значения: 0s-10s. По умолчанию: 5s. Поведение свойства: - *поддерживается*, если значение `max_attempts` больше 1.
timeout	string	Тайм-аут. Поддерживаются суффиксы времени (например, 30s, 1m). Поддерживаются пользовательские макросы. Возможные значения: 1s-60s. По умолчанию: 5s.
http_proxy	string	Строка подключения к прокси HTTP(S), заданная в формате *[protocol]://[username[:password]@]proxy.example.com[:port]*. Поддерживаются пользовательские макросы.

Property	Type	Description
authtype	integer	Метод аутентификации HTTP. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.
username	string	Имя пользователя. Поддерживаются пользовательские макросы. Поведение свойства: - *поддерживается*, если для `authtype` установлено значение «Basic», «NTLM», «Kerberos» или «Digest»
password	string	Пароль. Поддерживаются пользовательские макросы. Поведение свойства: - *поддерживается*, если для `authtype` установлено значение «Basic», «NTLM», «Kerberos» или «Digest»
token	string	Токен Bearer. Поддерживаются пользовательские макросы. Поведение свойства: - *обязательно*, если для `authtype` установлено значение «Bearer»
verify_peer	integer	Проверять ли подлинность сертификата узла сети. Возможные значения: 0 - Не проверять; 1 - *(по умолчанию)* Проверять.
verify_host	integer	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - Не проверять; 1 - *(по умолчанию)* Проверять.
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поддерживаются пользовательские макросы.
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поддерживаются пользовательские макросы.
ssl_key_password	string	Пароль для файла ключа SSL. Поддерживаются пользовательские макросы.
description	text	Описание коннектора.
status	integer	Включен ли коннектор. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.
tags_evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.

Фильтр тегов

Фильтр тегов позволяет экспортировать только совпадающие значения элементов данных или события. Если не задан, будет экспортироваться все. Объект фильтра тегов имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега. Property behavior: - *required*
operator	integer	Оператор условия. Возможные значения: 0 - *(default)* Равно; 1 - Не равно; 2 - Содержит; 3 - Не содержит; 12 - Существует; 13 - Не существует.
value	string	Значение тега.

connector.create

Описание

`object connector.create(object/array connectors)`

Этот метод позволяет создавать новые объекты коннектора.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Объекты коннектора для создания.

Помимо стандартных свойств коннектора, метод принимает следующие параметры.

Parameter	Type	Description
tags	array	Фильтр тегов коннектора.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных коннекторов в свойстве `connectorids`. Порядок возвращаемых идентификаторов соответствует порядку переданных коннекторов.

Примеры

Создание соединителя

Создайте соединитель для экспорта событий триггера с фильтром тегов. HTTP-аутентификация будет выполняться с использованием токена Bearer.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "connector.create",
    "params": [
        {
            "name": "Export of events",
            "data_type": 1,
            "url": "{$DATA_EXPORT_URL}",
            "authtype": 5,
            "token": "{$DATA_EXPORT_BEARER_TOKEN}",
            "tags": [
                {
                    "tag": "service",
```

```
                    "operator": 0,
                    "value": "mysqld"
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Источник

CConnector::create() в *ui/include/classes/api/services/CConnector.php*.

connector.delete

Описание

`object connector.delete(array connectorids)`

Этот метод позволяет удалять записи коннекторов.

> **Note:**
> Этот метод доступен только пользователю типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(array)` Идентификаторы коннекторов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных коннекторов в свойстве `connectorids`.

Примеры

Удаление нескольких соединителей

Удалить две записи соединителя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
```

```
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CConnector::delete() в *ui/include/classes/api/services/CConnector.php*.

connector.get

Описание

`integer/array connector.get(object parameters)`

Метод позволяет получать объекты коннекторов в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
connectorids	ID/array	Возвращать только коннекторы с указанными ID.
selectTags	query	Возвращать свойство `tags` с фильтром тегов коннектора. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение всех коннекторов

Получить все данные обо всех коннекторах и их свойствах.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
            "protocol": "0",
            "data_type": "1",
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
```

```
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```

Источник

CConnector:get() в *ui/include/classes/api/services/CConnector.php*.

connector.update

Описание

`object connector.update(object/array connectors)`

Этот метод позволяет обновлять существующие коннекторы.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства коннектора, которые необходимо обновить.

Для каждого коннектора должно быть определено свойство `connectorid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам коннектора, метод принимает следующие параметры.

Parameter	Type	Description
tags	array	Фильтр тегов коннектора для замены текущего фильтра тегов.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных коннекторов в свойстве `connectorids`.

Примеры

Изменение типа HTTP-аутентификации

Измените тип аутентификации HTTP на «Носитель» для соединителя с идентификатором «3».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
```

```
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Обновление фильтра тегов

Измените фильтр тегов для соединителя с идентификатором «5».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": [
        {
            "connectorid": 5,
            "tags_evaltype": 2,
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                },
                {
                    "tag": "error",
                    "operator": 12,
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CConnector::update() в *ui/include/classes/api/services/CConnector.php*.

Map

Этот класс предназначен для работы с картами.

Ссылки на объекты:

- Map
- Map element
  - Map element Host
  - Map element Host group
  - Map element Map
  - Map element Trigger
  - Map element Tag
  - Map element URL
- Map link
  - Map link trigger
  - Map link indicators
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

Доступные методы:

- map.create - создать новые карты
- map.delete - удалить карты
- map.get - получить карты
- map.update - обновить карты

Объект карты

Следующие объекты напрямую связаны с API `map`.

Карта

Объект карты имеет следующие свойства.

Property	Type	Description
sysmapid	ID	ID карты. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
height	integer	Высота карты в пикселях. Поведение свойства: - *обязательно* для операций создания
name	string	Имя карты. Поведение свойства: - *обязательно* для операций создания
width	integer	Ширина карты в пикселях. Поведение свойства: - *обязательно* для операций создания
backgroundid	ID	ID изображения, используемого в качестве фона карты.
background_scale	integer	Включать ли пропорциональное масштабирование фонового изображения. Возможные значения: 0 - отключить масштабирование; 1 - *(по умолчанию)* включить масштабирование изображения.

Property	Type	Description
expand_macros	integer	Включать ли раскрытие макросов в подписях при настройке карты. Возможные значения: 0 - *(по умолчанию)* не раскрывать макросы; 1 - раскрывать макросы.
expandproblem	integer	Будет ли отображаться триггер проблемы для элементов с одной проблемой. Возможные значения: 0 - всегда отображать количество проблем; 1 - *(по умолчанию)* отображать триггер проблемы, если проблема только одна.
grid_align	integer	Включать ли выравнивание по сетке. Возможные значения: 0 - отключить выравнивание по сетке; 1 - *(по умолчанию)* включить выравнивание по сетке.
grid_show	integer	Показывать ли сетку на карте. Возможные значения: 0 - не показывать сетку; 1 - *(по умолчанию)* показывать сетку.
grid_size	integer	Размер сетки карты в пикселях. Поддерживаемые значения: 20, 40, 50, 75 и 100. По умолчанию: 50.
highlight	integer	Включена ли подсветка значков. Возможные значения: 0 - подсветка отключена; 1 - *(по умолчанию)* подсветка включена.
iconmapid	ID	ID карты значков, используемой на карте.
label_format	integer	Включать ли расширенные подписи. Возможные значения: 0 - *(по умолчанию)* отключить расширенные подписи; 1 - включить расширенные подписи.
label_location	integer	Расположение подписи элемента карты. Возможные значения: 0 - *(по умолчанию)* снизу; 1 - слева; 2 - справа; 3 - сверху.
label_string_host	string	Пользовательская подпись для элементов узла сети. Поведение свойства: - *обязательно*, если `label_type_host` установлено в ”custom”
label_string_hostgroup	string	Пользовательская подпись для элементов группы узлов сети. Поведение свойства: - *обязательно*, если `label_type_hostgroup` установлено в ”custom”
label_string_image	string	Пользовательская подпись для элементов изображения. Поведение свойства: - *обязательно*, если `label_type_image` установлено в ”custom”

Property	Type	Description
label_string_map	string	Пользовательская подпись для элементов карты. Поведение свойства: - *обязательно*, если `label_type_map` установлено в ”custom”
label_string_trigger	string	Пользовательская подпись для элементов триггера. Поведение свойства: - *обязательно*, если `label_type_trigger` установлено в ”custom”
label_type	integer	Тип подписи элемента карты. Возможные значения: 0 - подпись; 1 - IP-адрес; 2 - *(по умолчанию)* имя элемента; 3 - только статус; 4 - ничего.
label_type_host	integer	Тип подписи для элементов узла сети. Возможные значения: 0 - подпись; 1 - IP-адрес; 2 - *(по умолчанию)* имя элемента; 3 - только статус; 4 - ничего; 5 - пользовательский.
label_type_hostgroup	integer	Тип подписи для элементов группы узлов сети. Возможные значения: 0 - подпись; 2 - *(по умолчанию)* имя элемента; 3 - только статус; 4 - ничего; 5 - пользовательский.
label_type_image	integer	Тип подписи для элементов группы узлов сети. Возможные значения: 0 - подпись; 2 - *(по умолчанию)* имя элемента; 4 - ничего; 5 - пользовательский.
label_type_map	integer	Тип подписи для элементов карты. Возможные значения: 0 - подпись; 2 - *(по умолчанию)* имя элемента; 3 - только статус; 4 - ничего; 5 - пользовательский.
label_type_trigger	integer	Тип подписи для элементов триггера. Возможные значения: 0 - подпись; 2 - *(по умолчанию)* имя элемента; 3 - только статус; 4 - ничего; 5 - пользовательский.
markelements	integer	Подсвечивать ли элементы карты, которые недавно изменили свой статус. Возможные значения: 0 - *(по умолчанию)* не подсвечивать элементы; 1 - подсвечивать элементы.

Property	Type	Description
severity_min	integer	Минимальная серьезность триггеров, которые будут отображаться на карте. См. свойство `severity` триггера для списка поддерживаемых уровней серьезности триггеров.
show_element_label	integer	Как по умолчанию отображать подписи элементов. Возможные значения: 0 - всегда отображать; 1 - *(по умолчанию)* автоматически скрывать.
show_link_label	integer	Как по умолчанию отображать подписи связей. Возможные значения: 0 - всегда отображать; 1 - *(по умолчанию)* автоматически скрывать.
show_unack	integer	Как должны отображаться проблемы. Возможные значения: 0 - *(по умолчанию)* отображать количество всех проблем; 1 - отображать только количество неподтвержденных проблем; 2 - отображать количество подтвержденных и неподтвержденных проблем отдельно.
userid	ID	ID пользователя, являющегося владельцем карты.
private	integer	Тип общего доступа к карте. Возможные значения: 0 - общедоступная карта; 1 - *(по умолчанию)* частная карта.
show_suppressed	integer	Показывать ли подавленные проблемы. Возможные значения: 0 - *(по умолчанию)* скрывать подавленные проблемы; 1 - показывать подавленные проблемы.

Элемент карты

Объект элемента карты определяет объект, отображаемый на карте. Он имеет следующие свойства.

Property	Type	Description
selementid	ID	ID элемента карты. Поведение свойства: - *только для чтения*
elements	array	Объект данных элемента. Поведение свойства: - *обязательно*, если `elementtype` установлен в ”host”, ”map”, ”trigger” или ”host group”
elementtype	integer	Тип элемента карты. Возможные значения: 0 - host; 1 - map; 2 - trigger; 3 - host group; 4 - image. Поведение свойства: - *обязательно*

Property	Type	Description
iconid_off	ID	ID изображения, используемого для отображения элемента в состоянии по умолчанию. Поведение свойства: - *обязательно*
areatype	integer	Как должны отображаться отдельные узлы сети группы узлов сети. Возможные значения: 0 - *(по умолчанию)* элемент группы узлов сети занимает всю карту; 1 - элемент группы узлов сети имеет фиксированный размер.
elementsubtype	integer	Как элемент группы узлов сети должен отображаться на карте. Возможные значения: 0 - *(по умолчанию)* отображать группу узлов сети как один элемент; 1 - отображать каждый узел сети в группе отдельно.
evaltype	integer	Метод оценки условия фильтрации тегов элемента карты evaluation method. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.
height	integer	Высота элемента группы узлов сети фиксированного размера в пикселях. По умолчанию: 200.
iconid_disabled	ID	ID изображения, используемого для отображения отключенных элементов карты. Поведение свойства: - *поддерживается*, если `elementtype` установлен в ”host”, ”map”, ”trigger” или ”host group”
iconid_maintenance	ID	ID изображения, используемого для отображения элементов карты в обслуживании. Поведение свойства: - *поддерживается*, если `elementtype` установлен в ”host”, ”map”, ”trigger” или ”host group”
iconid_on	ID	ID изображения, используемого для отображения элементов карты с проблемами. Поведение свойства: - *поддерживается*, если `elementtype` установлен в ”host”, ”map”, ”trigger” или ”host group”
label	string	Метка элемента.
label_location	integer	Расположение метки элемента карты. Возможные значения: -1 - *(по умолчанию)* расположение по умолчанию; 0 - снизу; 1 - слева; 2 - справа; 3 - сверху.
permission	integer	Тип уровня доступа. Возможные значения: -1 - нет; 2 - только чтение; 3 - чтение-запись.

Property	Type	Description
show_label	integer	Как отображать метку элемента. Возможные значения: -1 - *(по умолчанию)* значение карты по умолчанию; 0 - всегда отображать; 1 - автоматически скрывать.
sysmapid	ID	ID карты, к которой принадлежит элемент. Поведение свойства: - *только для чтения*
urls	array	URL-адреса элемента карты. Объект URL-адреса элемента карты подробно описан ниже.
use_iconmap	integer	Нужно ли использовать сопоставление значков для элементов узлов сети. Возможные значения: 0 - не использовать сопоставление значков; 1 - *(по умолчанию)* использовать сопоставление значков.
viewtype	integer	Алгоритм размещения элемента группы узлов сети. Возможные значения: 0 - *(по умолчанию)* сетка.
width	integer	Ширина элемента группы узлов сети фиксированного размера в пикселях. По умолчанию: 200.
x	integer	Координата X элемента в пикселях. По умолчанию: 0.
y	integer	Координата Y элемента в пикселях. По умолчанию: 0.
zindex	integer	Значение, используемое для упорядочивания элементов карты (z-index). По умолчанию: 0.

Объект узла сети элемента карты

Объект Host элемента карты определяет один элемент узла сети.

Property	Type	Description
hostid	ID	ID узла сети.

Элемент карты Группа узлов сети

Объект элемента карты Группа узлов сети определяет один элемент группы узлов сети.

Property	Type	Description
groupid	ID	ID группы узлов сети.

Элемент карты Map

Объект Map элемента карты определяет один элемент карты.

Property	Type	Description
sysmapid	ID	ID карты.

Объект триггера элемента карты

Объект Trigger элемента карты определяет один или несколько элементов триггера.

Property	Type	Description
triggerid	ID	ID триггера.

Тег элемента карты

Объект Tag элемента карты имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега элемента карты. Property behavior: - *required*
operator	integer	Оператор условия тега элемента карты. Возможные значения: 0 - *(default)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
value	string	Значение тега элемента карты.

URL элемента карты

Объект URL элемента карты определяет кликабельную ссылку, которая будет доступна для конкретного элемента карты. Он имеет следующие свойства:

Property	Type	Description
sysmapelementurlid	ID	ID URL элемента карты. Property behavior: - *read-only*
name	string	Подпись ссылки. Property behavior: - *required*
url	string	URL ссылки. Property behavior: - *required*
selementid	ID	ID элемента карты, которому принадлежит URL.

Связь карты

Объект связи карты определяет связь между двумя элементами карты. Он имеет следующие свойства.

Property	Type	Description
linkid	ID	ID связи карты. Поведение свойства: - *только для чтения*
sysmapid	ID	ID карты, к которой принадлежит связь.

Property	Type	Description
selementid1	ID	ID первого элемента карты, связанного на одном конце. Поведение свойства: - *обязательно*
selementid2	ID	ID первого элемента карты, связанного на другом конце. Поведение свойства: - *обязательно*
drawtype	integer	Стиль отображения линии связи. Возможные значения: 0 - *(по умолчанию)* линия; 2 - жирная линия; 3 - точечная линия; 4 - пунктирная линия.
color	string	Цвет линии в виде шестнадцатеричного кода цвета. По умолчанию: 000000.
label	string	Метка связи.
show_label	integer	Как отображать метку связи. Возможные значения: -1 - *(по умолчанию)* значение по умолчанию карты; 0 - всегда отображать; 1 - автоматически скрывать.
indicator_type	integer	Выберите тип индикатора связи. Возможные значения: 0 - *(по умолчанию)* статическая связь; 1 - триггер; 2 - значение элемента данных.
linktriggers	array	Триггеры связи карты, используемые в качестве индикаторов состояния связи. Объект триггера связи карты описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлен в ”trigger” (1).
itemid	ID	ID элемента данных. Поведение свойства: - *обязательно*, если `indicator_type` установлен в ”item value” (2).
highlights	array	Подсветки связи карты, используемые в качестве индикаторов состояния связи. Объект индикаторов связи карты описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлен в ”item value” (2).
thresholds	array	Пороги связи карты, используемые в качестве индикаторов состояния связи. Объект индикаторов связи карты описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлен в ”item value” (2).

Property	Type	Description
permission	integer	Тип уровня доступа. Возможные значения: -1 - нет; 2 - только чтение; 3 - чтение и запись.

Триггер ссылки карты

Объект триггера ссылки карты определяет индикатор состояния ссылки карты на основе состояния триггера. Он имеет следующие свойства:

Property	Type	Description
triggerid	ID	ID триггера, используемого в качестве индикатора ссылки. Property behavior: - *required*
color	string	Цвет индикатора в виде шестнадцатеричного кода цвета. По умолчанию: DD0000.
drawtype	integer	Стиль отображения индикатора. Возможные значения: 0 - *(по умолчанию)* линия; 2 - жирная линия; 3 - пунктирная линия; 4 - штриховая линия.

Индикаторы ссылок карты

Объект индикаторов ссылок карты определяет индикатор состояния ссылки карты на основе значения элемента данных. Он имеет следующие свойства:

Свойство	Type	Описание
pattern	string	Регулярное выражение для сравнения. Доступно только для выделений.
sortorder	integer	Используется для определения порядка сортировки выделений. Доступно только для выделений.
threshold	string	Числовое значение для сравнения. Доступно только для порогов.
drawtype	integer	Стиль отрисовки индикатора. Возможные значения: 0 - *(по умолчанию)* линия; 2 - жирная линия; 3 - пунктирная линия; 4 - штриховая линия.
color	string	Цвет индикатора в виде шестнадцатеричного кода цвета. По умолчанию: DD0000. Поведение свойства: - *обязательно*

URL карты

Объект URL карты определяет кликабельную ссылку, которая будет доступна для всех элементов определенного типа на карте. Он имеет следующие свойства:

Property	Type	Description
sysmapurlid	ID	ID URL карты. Property behavior: - *read-only*
name	string	Подпись ссылки. Property behavior: - *required*
url	string	URL ссылки. Property behavior: - *required*
elementtype	integer	Тип элемента карты, для которого будет доступен URL. Список поддерживаемых типов см. в свойстве `type` элемента карты. По умолчанию: 0.
sysmapid	ID	ID карты, которой принадлежит URL.

Пользователь карты

Список прав доступа к карте на основе пользователей. Он имеет следующие свойства:

Property	Type	Description
sysmapuserid	ID	ID пользователя карты. Property behavior: - *read-only*
userid	ID	ID пользователя. Property behavior: - *required*
permission	integer	Тип уровня доступа. Возможные значения: 2 - только чтение; 3 - чтение и запись. Property behavior: - *required*

Группа пользователей карты

Список прав доступа к карте на основе групп пользователей. Имеет следующие свойства:

Property	Type	Description
sysmapusrgrpid	ID	ID группы пользователей карты. Property behavior: - *read-only*
usrgrpid	ID	ID группы пользователей. Property behavior: - *required*

Property	Type	Description
permission	integer	Тип уровня доступа. Возможные значения: 2 - только чтение; 3 - чтение-запись. Property behavior: - *required*

Фигуры карты

Объект фигуры карты определяет геометрическую фигуру (с текстом или без него), отображаемую на карте. Он имеет следующие свойства:

Property	Type	Description
sysmap_shapeid	ID	ID элемента фигуры карты. Поведение свойства: - *только для чтения*
type	integer	Тип элемента фигуры карты. Возможные значения: 0 - прямоугольник; 1 - эллипс. Свойство обязательно при создании новых фигур. Поведение свойства: - *обязательное*
x	integer	Координата X фигуры в пикселях. По умолчанию: 0.
y	integer	Координата Y фигуры в пикселях. По умолчанию: 0.
width	integer	Ширина фигуры в пикселях. По умолчанию: 200.
height	integer	Высота фигуры в пикселях. По умолчанию: 200.
text	string	Текст фигуры.
font	integer	Шрифт текста внутри фигуры. Возможные значения: 0 - Georgia, serif 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif 2 - ”Times New Roman”, Times, serif 3 - Arial, Helvetica, sans-serif 4 - ”Arial Black”, Gadget, sans-serif 5 - ”Comic Sans MS”, cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif 8 - Tahoma, Geneva, sans-serif 9 - ”Trebuchet MS”, Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - ”Courier New”, Courier, monospace 12 - ”Lucida Console”, Monaco, monospace По умолчанию: 9.

Property	Type	Description
font_size	integer	Размер шрифта в пикселях. По умолчанию: 11.
font_color	string	Цвет шрифта. По умолчанию: 000000.
text_halign	integer	Горизонтальное выравнивание текста. Возможные значения: 0 - по центру; 1 - слева; 2 - справа. По умолчанию: 0.
text_valign	integer	Вертикальное выравнивание текста. Возможные значения: 0 - по середине; 1 - сверху; 2 - снизу. По умолчанию: 0.
border_type	integer	Тип границы. Возможные значения: 0 - нет; 1 - ——————; 2 - ---; 3 - – – –. По умолчанию: 0.
border_width	integer	Ширина границы в пикселях. По умолчанию: 0.
border_color	string	Цвет границы. По умолчанию: 000000.
background_color	string	Цвет фона (цвет заливки). По умолчанию: `(empty)`.
zindex	integer	Значение, используемое для упорядочивания всех фигур и линий (z-index). По умолчанию: 0.

Линии карты

Объект линии карты определяет линию, отображаемую на карте. Он имеет следующие свойства:

Property	Type	Description
sysmap_shapeid	ID	ID элемента фигуры карты. Поведение свойства: *- только для чтения*
x1	integer	Координата X точки 1 линии в пикселях. По умолчанию: 0.
y1	integer	Координата Y точки 1 линии в пикселях. По умолчанию: 0.

Property	Type	Description
x2	integer	Координата X точки 2 линии в пикселях. По умолчанию: 200.
y2	integer	Координата Y точки 2 линии в пикселях. По умолчанию: 200.
line_type	integer	Тип линии. Возможные значения: 0 - нет; 1 - ——————; 2 - ┅┅; 3 - – – –. По умолчанию: 0.
line_width	integer	Ширина линии в пикселях. По умолчанию: 0.
line_color	string	Цвет линии. По умолчанию: 000000.
zindex	integer	Значение, используемое для упорядочивания всех фигур и линий (z-index). По умолчанию: 0.

map.create

Описание

`object map.create(object/array maps)`

Этот метод позволяет создавать новые карты.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Карты для создания.

В дополнение к стандартным свойствам карты, метод принимает следующие параметры.

Параметр	Тип	Описание
links	array	Связи карты для создания на карте.
selements	array	Элементы карты для создания на карте.
urls	array	URL-адреса карты для создания на карте.
users	array	Общий доступ пользователей карты для создания на карте.
userGroups	array	Общий доступ групп пользователей карты для создания на карте.
shapes	array	Фигуры карты для создания на карте.
lines	array	Линии карты для создания на карте.

> **Note:**
> Чтобы создать связи карты, вам нужно задать для элемента карты `selementid` произвольное значение, а затем использовать это значение для ссылки на этот элемент в свойствах `selementid1` или `selementid2` связей. Когда элемент будет создан, это значение будет заменено на корректный ID, сгенерированный Zabbix. См. пример.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы созданных карт под свойством sysmapids. Порядок возвращаемых идентификаторов соответствует порядку переданных карт.

Примеры

Создание пустой карты

Создайте карту без элементов.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Создание карты узлов сети

Создайте карту с двумя элементами узлов сети и связью между ними. Обратите внимание на использование временных значений "selementid1" и "selementid2" в объекте связи карты для ссылки на элементы карты.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "selementid": "1",
                "elements": [
                    {"hostid": "1033"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            },

            {
                "selementid": "2",
                "elements": [
                    {"hostid": "1037"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            }
        ],
        "links": [
```

```
            {
                "selementid1": "1",
                "selementid2": "2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Создание карты триггеров

Создайте карту с элементом триггера, который содержит два триггера.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Trigger map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "elements": [
                    {"triggerid": "12345"},
                    {"triggerid": "67890"}
                ],
                "elementtype": 2,
                "iconid_off": "2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Общий доступ к карте

Создайте карту с двумя типами общего доступа (пользователь и группа пользователей).

Запрос:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "map.create",
    "params": {
        "name": "Map sharing",
        "width": 600,
        "height": 600,
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Фигуры карты

Создайте карту с заголовком имени карты.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "shapes": [
            {
                "type": 0,
                "x": 0,
                "y": 0,
                "width": 600,
                "height": 11,
                "text": "{MAP.NAME}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
```

```
            "10"
        ]
    },
    "id": 1
}
```

Линии карты

Создайте линию карты.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map API lines",
        "width": 500,
        "height": 500,
        "lines": [
            {
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

См. также

- Элемент карты
- Ссылка карты
- URL карты
- Пользователь карты
- Группа пользователей карты
- Фигура карты
- Линия карты

Источник

CMap::create() в *ui/include/classes/api/services/CMap.php*.

map.delete

Описание

`object map.delete(array mapIds)`

Этот метод позволяет удалять карты.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы карт для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных карт под свойством `sysmapids`.

Примеры

Удаление нескольких карт

Удаление двух карт.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Источник

CMap::delete() в *ui/include/classes/api/services/CMap.php*.

map.get

Описание

`integer/array map.get(object parameters)`

Метод позволяет получать карты в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
sysmapids	ID/array	Возвращает только карты с указанными ID.
userids	ID/array	Возвращает только карты, принадлежащие указанным ID пользователей.

Parameter	Type	Description
expandUrls	flag	Добавляет глобальные URL карты к соответствующим элементам карты и разворачивает макросы во всех URL элементов карты.
selectIconMap	query	Возвращает свойство `iconmap` с картой значков, используемой на карте.
selectLinks	query	Возвращает свойство `links` со связями карты между элементами.
selectSelements	query	Возвращает свойство `selements` с элементами карты.
selectUrls	query	Возвращает свойство `urls` с URL карты.
selectUsers	query	Возвращает свойство `users` с пользователями, с которыми карта расшарена.
selectUserGroups	query	Возвращает свойство `userGroups` с группами пользователей, с которыми карта расшарена.
selectShapes	query	Возвращает свойство `shapes` с фигурами карты.
selectLines	query	Возвращает свойство `lines` с линиями карты.
sortfield	string/array	Сортирует результат по указанным свойствам. Возможные значения: `name`, `width`, `height`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение карты

Получить все данные о карте "3".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.get",
    "params": {
        "output": "extend",
        "selectSelements": "extend",
        "selectLinks": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "selectShapes": "extend",
        "selectLines": "extend",
        "sysmapids": "3"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "0",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": []
                },
                {
                    "selementid": "12",
                    "sysmapid": "3",
                    "elementtype": "0",
```

```
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
```

```
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
        "grid_show": "1",
        "grid_align": "1",
        "label_format": "0",
        "label_type_host": "2",
        "label_type_hostgroup": "2",
        "label_type_trigger": "2",
        "label_type_map": "2",
        "label_type_image": "2",
        "label_string_host": "",
        "label_string_hostgroup": "",
        "label_string_trigger": "",
        "label_string_map": "",
        "label_string_image": "",
        "iconmapid": "0",
        "expand_macros": "0",
        "severity_min": "0",
        "userid": "1",
        "private": "1",
```

```
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```

См. также

- Карта значков
- Элемент карты
- Ссылка карты
- URL карты
- Пользователь карты
- Группа пользователей карты
- Фигуры карты
- Линии карты

Источник

CMap::get() в *ui/include/classes/api/services/CMap.php*.

map.update

Описание

`object map.update(object/array maps)`

Этот метод позволяет обновлять существующие карты.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Сопоставление свойств, которые необходимо обновить.

Свойство `mapid` должно быть определено для каждой карты, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам карты, метод принимает следующие параметры.

Parameter	Type	Description
links	array	Ссылки карты для замены существующих ссылок.
selements	array	Элементы карты для замены существующих элементов.
urls	array	URL карты для замены существующих URL.
users	array	Пользователи карты для замены существующих элементов.
userGroups	array	Группы пользователей карты для замены существующих элементов.
shapes	array	Фигуры карты для замены существующих фигур.
lines	array	Линии карты для замены существующих линий.

> **Note:**
> Чтобы создать ссылки карты между новыми элементами карты, вам нужно задать для `selementid` элемента произвольное значение, а затем использовать это значение для ссылки на данный элемент в свойствах `selementid1` или `selementid2` ссылки. Когда элемент будет создан, это значение будет заменено на корректный ID, сгенерированный Zabbix. См. пример для map.create.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных карт под свойством `sysmapids`.

Примеры

Изменение размера карты

Измените размер карты до 1200x1200 пикселей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Изменение владельца карты

Доступно только администраторам и супер-администраторам.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

См. также

- Элемент карты
- Ссылка карты
- URL карты
- Пользователь карты
- Группа пользователей карты
- Фигуры карты
- Линии карты

Источник

CMap::update() в *ui/include/classes/api/services/CMap.php*.

MFA

Этот класс предназначен для работы с методами MFA (Multi-Factor Authentication).

Ссылки на объекты:

- MFA

Доступные методы:

- mfa.create - создать новые методы MFA
- mfa.delete - удалить методы MFA
- mfa.get - получить методы MFA
- mfa.update - обновить методы MFA

Объект MFA

Следующие объекты напрямую связаны с API `mfa`.

MFA

Объект MFA (Multi-Factor Authentication) имеет следующие свойства.

Property	Type	Description
mfaid	ID	ID метода MFA. Property behavior: - *только для чтения* - *обязательно* для операций обновления
type	integer	Тип метода MFA. Возможные значения: 1 - TOTP (одноразовые пароли на основе времени); 2 - Duo Universal Prompt.
name	string	Уникальное имя метода MFA. Property behavior: - *обязательно* для операций создания
hash_function	integer	Тип хэш-функции для генерации кодов TOTP. Возможные значения: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Property behavior: - *обязательно* если `type` установлено в ”TOTP”
code_length	integer	Длина кода подтверждения. Возможные значения: 6 - 6-значный; 8 - 8-значный. Property behavior: - *обязательно* если `type` установлено в ”TOTP”
api_hostname	string	Имя хоста API, предоставленное службой аутентификации Duo. Property behavior: - *обязательно* если `type` установлено в ”Duo Universal Prompt”
clientid	string	ID клиента, предоставленный службой аутентификации Duo. Property behavior: - *обязательно* если `type` установлено в ”Duo Universal Prompt”

Property	Type	Description
client_secret	string	Секрет клиента, предоставленный службой аутентификации Duo. Property behavior: - *только для записи* - *обязательно* если `type` установлено в "Duo Universal Prompt"

mfa.create

Описание

`object mfa.create(object/array MFA methods)`

Этот метод позволяет создавать новые методы MFA.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` MFA-методы для создания.

Метод принимает MFA-методы со стандартными свойствами MFA-метода.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных методов MFA в свойстве `mfaids`. Порядок возвращаемых идентификаторов соответствует порядку переданных элементов.

Примеры

Создание методов MFA

Создайте метод MFA "Zabbix TOTP", использующий одноразовые пароли на основе времени (TOTP), с функцией хеширования для генерации кодов TOTP, установленной в SHA-1, и длиной кода проверки, установленной в 6 цифр.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CMfa::create() в *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Описание

`object mfa.delete(array mfaids)`

Этот метод позволяет удалять методы MFA.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(array)` Идентификаторы методов MFA, которые нужно удалить.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных методов MFA в свойстве `mfaids`.

Примеры

Удаление методов MFA

Удалить метод MFA.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

Исходный код

CMfa::delete() в *ui/include/classes/api/services/CMfa.php*.

mfa.get

Описание

`integer/array mfa.get(object parameters)`

Метод позволяет получить методы MFA в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
mfaids	ID/array	Возвращать только методы MFA с указанными ID.
selectUsrgrps	query	Возвращать свойство `usrgrps` с группами пользователей, связанными с методами MFA. Поддерживает `count`.
filter	object	Возвращать только те результаты, которые точно соответствуют заданному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сопоставления. Поддерживаемые свойства: `mfaid` - ID метода MFA; `type` - Тип метода MFA.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `name`.
search	object	Возвращать результаты, соответствующие заданному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств, а значения — строки для поиска. Если дополнительные параметры не заданы, будет выполнен поиск `LIKE "%…%"`. Поддерживаемые свойства: `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Поиск методов MFA по имени

Получить все методы MFA, в названии которых есть ”Zabbix”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```


Исходный код

CMfa::get() в *ui/include/classes/api/services/CMfa.php*.

mfa.update

Описание

`object mfa.update(object/array MFA methods)`

Этот метод позволяет обновлять существующие методы MFA.

> **Note:**
> Этот метод доступен только для пользователей типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства метода MFA, которые необходимо обновить.

Для каждого элемента должно быть определено свойство `mfaid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Метод принимает методы MFA со стандартными свойствами метода MFA.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных методов MFA в свойстве `mfaids`.

Примеры

Обновление свойств метода

Обновите функцию хеширования для генерации кодов TOTP и длину кода проверки для метода MFA "Zabbix TOTP", использующего одноразовые пароли на основе времени (TOTP).

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
```

```
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CMfa::update() в *ui/include/classes/api/services/CMfa.php*.

SLA

Этот класс предназначен для работы с объектами SLA (Service Level Agreement), используемыми для оценки производительности ИТ-инфраструктуры и бизнес-сервисов.

Ссылки на объекты:

- SLA
- Расписание SLA
- Исключенное время простоя SLA
- Тег сервиса SLA

Доступные методы:

- sla.create - создать новые SLA
- sla.delete - удалить SLA
- sla.get - получить SLA
- sla.getsli - получить данные индикатора уровня сервиса (SLI) для SLA
- sla.update - обновить SLA

Объект SLA

Следующие объекты напрямую связаны с API `sla` (Service Level Agreement).

SLA

Объект SLA имеет следующие свойства.

Property	Type	Description
slaid	ID	ID SLA. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя SLA. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
period	integer	Период отчетности SLA. Возможные значения: 0 - ежедневно; 1 - еженедельно; 2 - ежемесячно; 3 - ежеквартально; 4 - ежегодно. Поведение свойства: - *обязательно* для операций создания
slo	float	Минимально допустимая цель уровня обслуживания, выраженная в процентах. Если показатель уровня обслуживания (SLI) опускается ниже, SLA считается находящимся в состоянии проблемы/невыполненным. Возможные значения: 0-100 (до 4 знаков после запятой). Поведение свойства: - *обязательно* для операций создания
effective_date	integer	Дата вступления SLA в силу. Возможные значения: временная метка даты в UTC.
timezone	string	Часовой пояс отчетности, например: `Europe/London`, `UTC`. Полный список поддерживаемых часовых поясов см. в документации PHP. Поведение свойства: - *обязательно* для операций создания
status	integer	Статус SLA. Возможные значения: 0 - *(по умолчанию)* отключенный SLA; 1 - включенный SLA.
description	string	Описание SLA.

Расписание SLA

Объект расписания SLA определяет периоды, в течение которых подключенные сервисы должны находиться в рабочем состоянии. Он имеет следующие свойства.

Property	Type	Description
period_from	integer	Время начала повторяющегося недельного периода времени (включительно). Возможные значения: число секунд (отсчет с воскресенья). Поведение свойства: - *required*
period_to	integer	Время окончания повторяющегося недельного периода времени (исключительно). Возможные значения: число секунд (отсчет с воскресенья). Поведение свойства: - *required*

Исключенное время простоя SLA

Объект исключенного времени простоя определяет периоды, в которые для подключенного сервиса(ов) запланировано состояние недоступности, при этом это не влияет на SLI, например, во время планового обслуживания. Он имеет следующие свойства.

Property	Type	Description
name	string	Имя исключенного периода простоя. Поведение свойства: - *required*
period_from	integer	Время начала исключенного периода простоя (включительно). Possible values: timestamp. Поведение свойства: - *required*
period_to	integer	Время окончания исключенного периода простоя (не включая). Possible values: timestamp. Поведение свойства: - *required*

Тег службы SLA

Объект тега службы SLA связывает службы, которые нужно включить в расчеты SLA. Он имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега SLA-сервиса. Поведение свойства: - *required*
operator	integer	Оператор тега SLA-сервиса. Возможные значения: 0 - *(default)* Равно; 2 - Содержит.
value	string	Значение тега SLA-сервиса.

sla.create

Описание

`object sla.create(object/array SLAs)`

Этот метод позволяет создавать новые объекты SLA.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Объекты SLA для создания.

Помимо стандартных свойств SLA, метод принимает следующие параметры.

Параметр	Тип	Описание
service_tags	array	Теги служб SLA, которые будут созданы для SLA. Поведение параметра: - *required*

Параметр	Тип	Описание
schedule	array	Расписание SLA, которое будет создано для SLA. Указание пустого параметра будет интерпретировано как расписание 24x7. По умолчанию: расписание 24x7.
excluded_downtimes	array	Исключенные периоды простоя SLA, которые будут созданы для SLA.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы созданных SLA в свойстве slaids. Порядок возвращаемых идентификаторов соответствует порядку переданных SLA.

Примеры

Создание SLA

Укажите создание записи SLA для: * отслеживания времени безотказной работы служб, связанных с SQL-движком; * пользовательского расписания на все будние дни, за исключением последнего часа в субботу; * даты вступления в силу — последний день 2022 года; * запланированного простоя длительностью 1 час 15 минут, начиная с полуночи 4 июля; * включенного еженедельного расчета отчета SLA; * минимально допустимого SLO на уровне 99.9995%.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Database Uptime",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Provide excellent uptime for main database engines.",
            "effective_date": 1672444800,
            "status": 1,
            "schedule": [
                {
                    "period_from": 0,
                    "period_to": 601200
                }
            ],
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CSla::create() в *ui/include/classes/api/services/CSla.php*.

sla.delete

Описание

`object sla.delete(array slaids)`

Этот метод позволяет удалять записи SLA.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы SLA, которые нужно удалить.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных SLA в свойстве `slaids`.

Примеры

Удаление нескольких SLA

Удалите две записи SLA.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CSla::delete() в *ui/include/classes/api/services/CSla.php*.

sla.get

Описание

`integer/array sla.get(object parameters)`

Метод позволяет получать объекты SLA в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
slaids	ID/array	Возвращать только SLA с указанными ID.
serviceids	ID/array	Возвращать только SLA, соответствующие указанным сервисам.
selectSchedule	query	Возвращать свойство `schedule` с расписаниями SLA. Поддерживает `count`.
selectExcludedDowntimes	query	Возвращать свойство `excluded_downtimes` с исключенными периодами простоя SLA. Поддерживает `count`.
selectServiceTags	query	Возвращать свойство `service_tags` с тегами сервисов SLA. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использован параметр `countOutput`.

Примеры

Получение всех SLA

Получите все данные обо всех SLA и их свойствах.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.get",
```

```
    "params": {
        "output": "extend",
        "selectSchedule": ["period_from", "period_to"],
        "selectExcludedDowntimes": ["name", "period_from", "period_to"],
        "selectServiceTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    },
    "id": 1
}
```

Источник

CSla:get() в *ui/include/classes/api/services/CSla.php*.

sla.getsli

Описание

`object sla.getsli(object parameters)`

Этот метод позволяет вычислять данные индикатора уровня сервиса (SLI) для соглашения об уровне сервиса (SLA).

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, содержащие SLA ID, отчетные периоды и, при необходимости, ID сервисов, для которых вычисляется SLI.

Parameter	Type	Description
slaid	ID	ID SLA, для которого нужно вернуть информацию о доступности. Поведение параметра: *- required*
period_from	timestamp	Начальная временная метка (включительно) для отчета SLI. Возможные значения: Unix timestamp.
period_to	timestamp	Конечная временная метка (включительно) для отчета SLI. Возможные значения: Unix timestamp.
periods	integer	Количество периодов для отчета. Возможные значения: 1-100
serviceids	ID/array	ID сервисов, для которых нужно вернуть SLI.

Разбиение периодов

Следующая таблица демонстрирует порядок возвращаемых фрагментов периодов в зависимости от комбинаций параметров.

Note:
Возвращаемые периоды не предшествуют первому доступному периоду (на основе даты вступления SLA в силу) и не выходят за пределы текущего периода.

Parameters			Returned periods
period_from	**period_to**	**periods**	
-	-	Последние 20 периодов, включая текущий.	
-	-	N	Последние N периодов.
-	specified	-	Последние 20 периодов перед `period_to`.
-	specified	N	Последние N периодов перед `period_to`.
specified	-	-	Первые 20 периодов, начиная с `period_from`.
specified	-	N	Первые N периодов, начиная с `period_from`.
specified	specified	-	До 100 периодов в указанном диапазоне.
specified	specified	N	N периодов в указанном диапазоне.

Возвращаемые значения

`(object)` Возвращает результаты вычисления.

Property	Type	Description
periods	array	Список отчетных периодов. Каждый отчетный период представлен объектом, состоящим из: - `period_from` - (`timestamp`) Начальная метка времени (включительно) отчетного периода. - `period_to` - (`timestamp`) Конечная метка времени (исключительно) отчетного периода. Периоды отсортированы по `period_from`, при этом более ранние периоды идут первыми.
serviceids	array	Список идентификаторов сервисов в отчетных периодах. Порядок сортировки списка не определен. Даже если параметр `serviceids` был передан методу `sla.getsli`.
sli	array	Данные SLI (в виде **двумерного массива**) для каждого отчетного периода и сервиса. Индекс свойства `periods` используется как **первое** измерение свойства `sli`. Индекс свойства `serviceids` используется как **второе** измерение свойства `sli`.

Данные SLI

Данные SLI, возвращаемые для каждого указанного периода и сервиса, состоят из:

Property	Type	Description
uptime	integer	Время, в течение которого сервис находился в состоянии *OK* в течение запланированного времени работы, за вычетом исключенных простоев.
downtime	integer	Время, в течение которого сервис находился в состоянии *not OK* в течение запланированного времени работы, за вычетом исключенных простоев.
sli	float	SLI (в процентах от общего времени работы), основанный на uptime и downtime.
error_budget	integer	Бюджет ошибок (в секундах), основанный на SLI и SLO.

Property	Type	Description
excluded_downtimes	array	Массив исключенных простоев в этом отчетном периоде. Каждый объект будет содержать следующие параметры: - `name` - (`string`) Имя исключенного простоя. - `period_from` - (`timestamp`) Начальная метка времени (включительно) исключенного простоя. - `period_to` - (`timestamp`) Конечная метка времени (исключительно) исключенного простоя. Исключенные простои сортируются по `period_from`, при этом более ранние периоды идут первыми.

Примеры

Расчет SLI для ежедневного SLA

Получите данные SLI для услуг с ID "1" и "4", которые связаны с SLA с ID "1". Получите данные для одного периода до "1761861599" (Oct 30 2025 23:59:59 GMT+0200). Поскольку период отчетности SLA является ежедневным, данные SLI извлекаются с "1761775200" (Oct 30 2025 00:00:00 GMT+0200) по "1761861600" (Oct 31 2025 00:00:00 GMT+0200).

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
```

```
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
                            "period_from": 1761825600,
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```

Расчет SLI для ежемесячного SLA

Получите данные SLI по услугам с ID ”50”, ”60” и ”70”, которые связаны с SLA с ID ”5”. Получите данные за три периода, начиная с ”1635724800” (Nov 01 2021 00:00:00 UTC). Поскольку период отчетности SLA является ежемесячным, данные SLI извлекаются за следующие три месяца:

- С ”1635724800” (Nov 01 2021 00:00:00 UTC) по ”1638316800” (Dec 01 2021 00:00:00 UTC)
- С ”1638316800” (Dec 01 2021 00:00:00 UTC) по ”1640995200” (Jan 01 2022 00:00:00 UTC)
- С ”1640995200” (Jan 01 2022 00:00:00 UTC) по ”1643673600” (Feb 01 2022 00:00:00 UTC)

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
```

```
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
```

```
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


Источник

CSla::getSli() в *ui/include/classes/api/services/CSla.php*

sla.update

Описание

`object sla.update(object/array slaids)`

Этот метод позволяет обновлять существующие записи SLA.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства SLA, которые необходимо обновить.

Свойство `slaid` должно быть определено для каждого SLA, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам SLA, метод принимает следующие параметры.

Parameter	Type	Description
service_tags	array	Теги службы SLA для замены текущих тегов службы SLA.
schedule	array	Расписание SLA для замены текущего расписания. Если параметр указан пустым, это будет интерпретировано как расписание 24x7.
excluded_downtimes	array	Исключенные периоды простоя SLA для замены текущих.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных SLA в свойстве `slaids`.

Примеры

Обновление тегов сервиса

Сделать так, чтобы SLA с ID "5" рассчитывалось с ежемесячными интервалами для сервисов, связанных с NoSQL, без изменения его расписания или исключенных простоев; установить SLO на 95%.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": [
        {
            "slaid": "5",
            "name": "NoSQL Database engines",
            "slo": "95",
            "period": 2,
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "redis"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mongodb"
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Изменение расписания SLA

Переключите SLA с ID ”5” на расписание 24x7.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Изменение исключенных простоев для SLA

Добавьте запланированный простой на 4 часа для обновления RAM 6 апреля 2022 года, при этом сохранив (нужно определить заново) ранее существовавшее запланированное обновление программного обеспечения на 4 июля для SLA с ID ”5”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "excluded_downtimes": [
            {
                "name": "Software version upgrade rollout",
                "period_from": "1648760400",
                "period_to": "1648764900"
            },
            {
                "name": "RAM upgrade",
                "period_from": "1649192400",
                "period_to": "1649206800"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CSla::update() в *ui/include/classes/api/services/CSla.php*.

Token

Этот класс предназначен для работы с токенами.

Ссылки на объекты:

- Token

Доступные методы:

- token.create - создать новые токены
- token.delete - удалить токены
- token.get - получить токены
- token.update - обновить токены
- token.generate - сгенерировать токены

Объект token

Следующие объекты напрямую связаны с API `token`.

Токен

Объект token имеет следующие свойства.

Property	Type	Description
tokenid	ID	ID токена. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя токена. Поведение свойства: - *обязательно* для операций создания
description	text	Описание токена.
userid	ID	ID пользователя, которому назначен токен. По умолчанию: *current user*. Поведение свойства: - *константа*
lastaccess	timestamp	Дата и время последней аутентификации токена. ”0”, если токен ни разу не проходил аутентификацию. Поведение свойства: - *только для чтения*

Property	Type	Description
status	integer	Статус токена. Возможные значения: 0 - *(по умолчанию)* включенный токен; 1 - отключенный токен.
expires_at	timestamp	Дата и время истечения срока действия токена. ”0” для токенов без срока действия.
created_at	timestamp	Дата и время создания токена. Поведение свойства: - *только для чтения*
creator_userid	ID	ID пользователя, создавшего токен. Поведение свойства: - *только для чтения*

token.create

Описание

`object token.create(object/array tokens)`

Этот метод позволяет создавать новые токены.

Note:
Для роли пользователя требуется разрешение *Manage API tokens*, чтобы управлять токенами других пользователей.

Attention:
Токен, созданный этим методом, также должен быть сгенерирован перед тем, как его можно будет использовать.

Параметры

(/) Создаваемые токены.

Этот метод принимает токены со стандартными свойствами токена.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных токенов, указанных в свойстве `tokenids`. Порядок возвращаемых ID совпадает с порядком переданных токенов.

Примеры

Создание токена

Создайте включенный токен, срок действия которого никогда не истекает, и который выполняет аутентификацию пользователя с ID 2.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

Создайте отключенный токен, срок действия которого истекает 21 января 2021 года. Этот токен будет выполнять аутентификацию текущего пользователя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

Источник

CToken::create() в *ui/include/classes/api/services/CToken.php*.

token.delete

Описание

`object token.delete(array tokenids)`

Этот метод позволяет удалять токены.

> **Note:**
> Для роли пользователя требуется разрешение *Manage API tokens*, чтобы управлять токенами других пользователей.

Parameters

`(array)` ID токенов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых токенов, указанных в свойстве `tokenids`.

Примеры

Удаление нескольких токенов

Удалить два токена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Источник

CToken::delete() в *ui/include/classes/api/services/CToken.php*.

token.generate

Описание

`object token.generate(array tokenids)`

Этот метод позволяет генерировать токены.

Note:
Для роли пользователя требуется разрешение *Manage API tokens*, чтобы управлять токенами других пользователей.

Attention:
Токен может быть сгенерирован этим методом только в том случае, если он был создан.

Параметры

( ) ID токенов для генерации.

Возвращаемые значения

`(array)` Возвращает массив объектов, содержащих ID сгенерированного токена в свойстве `tokenid` и сгенерированную строку авторизации в свойстве `token`.

Property	Type	Description
tokenid	ID	ID токена.
token	string	Сгенерированная строка авторизации для этого токена.

Примеры

Создание нескольких токенов

Создайте два токена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Источник

CToken::generate() в *ui/include/classes/api/services/CToken.php*.

token.get

Описание

`integer/array token.get(object parameters)`

Метод позволяет получать токены в соответствии с заданными параметрами.

> **Note:**
> Только пользователям типа *Super admin* разрешено просматривать токены других пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
tokenids	ID/array	Возвращать только токены с указанными ID.
userids	ID/array	Возвращать только токены, созданные для указанных пользователей.
token	string	Возвращать только токены, созданные для указанного *Auth token*.
valid_at	timestamp	Возвращать только токены, которые действительны (не истекли) на указанную дату и время.
expired_at	timestamp	Возвращать только токены, срок действия которых истек (они недействительны) на указанную дату и время.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение токена

Получить все данные для токена с ID "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
    ],
    "id": 1
}
```

Источник

CToken::get() в *ui/include/classes/api/services/CToken.php*.

token.update

Описание

`object token.update(object/array tokens)`

Этот метод позволяет обновлять существующие токены.

Note:
Для роли пользователя требуется разрешение *Manage API tokens*, чтобы управлять токенами других пользователей.

Параметры

`(object/array)` Свойства токена, которые необходимо обновить.

Свойство `tokenid` должно быть определено для каждого токена, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Метод принимает токены со стандартными свойствами токена.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых токенов, указанных в свойстве `tokenids`.

Примеры

Удаление срока действия токена

Удалите дату истечения срока действия токена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

Источник

CToken::update() в *ui/include/classes/api/services/CToken.php*.

User

Этот класс предназначен для работы с пользователями.

Ссылки на объекты:

- User
- Media

Доступные методы:

- user.checkauthentication - проверка и продление сеансов пользователей
- user.create - создание новых пользователей
- user.delete - удаление пользователей
- user.get - получение пользователей
- user.login - вход в API
- user.logout - выход из API
- user.provision - подготовка пользователей LDAP
- user.resettotp - сброс секретов TOTP пользователей

- user.unblock - разблокировка пользователей
- user.update - обновление пользователей

Объект user

Следующие объекты напрямую связаны с API `user`.

Пользователь

Объект пользователя имеет следующие свойства.

Property	Type	Description
userid	ID	Идентификатор пользователя. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
username	string	Имя пользователя. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для подготовленных пользователей, если пользователь связан с каталогом пользователей (`userdirectoryid` задано допустимым значением, отличным от ”0”), включено предоставление пользователей из каталога (`provision_status` объекта каталога пользователей имеет значение ”1”), а также включена аутентификация для всех настроенных LDAP- или SAML-провижининговов (`ldap_jit_status` объекта аутентификации имеет значение ”Enabled for configured LDAP IdPs” или `saml_jit_status` объекта аутентификации имеет значение ”Enabled for configured SAML IdPs”)
passwd	string	Пароль пользователя. Значение этого параметра может быть пустой строкой, если пользователь связан с каталогом пользователей. Поведение свойства: - *только для записи*
roleid	ID	Идентификатор роли пользователя. Обратите внимание, что пользователи без роли могут войти в Zabbix только с использованием аутентификации LDAP или SAML, если их данные LDAP/SAML соответствуют сопоставлениям групп пользователей, настроенным в Zabbix.
attempt_clock	timestamp	Время последней неудачной попытки входа. Поведение свойства: - *только для чтения*
attempt_failed	integer	Количество недавних неудачных попыток входа. Поведение свойства: - *только для чтения*
attempt_ip	string	IP-адрес, с которого была выполнена последняя неудачная попытка входа. Поведение свойства: - *только для чтения*
autologin	integer	Определяет, включен ли автоматический вход. Возможные значения: 0 - *(по умолчанию)* автоматический вход отключен; 1 - автоматический вход включен.

Property	Type	Description
autologout	string	Время жизни сеанса пользователя. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h). Если задано значение 0s, срок действия сеанса никогда не истекает. По умолчанию: 15m.
default_maintenance_period	string	Период обслуживания по умолчанию для обслуживания, созданного с помощью ярлыка. Принимает количество секунд или единицу времени с суффиксом (300s, 5m, 1h, 1d, 1w). Указанное значение округляется вниз до целых минут. По умолчанию: 1h.
lang	string	Код языка пользователя, например, en_US. По умолчанию: default - системный язык по умолчанию.
name	string	Имя пользователя.
provisioned	integer	Определяет, был ли пользователь подготовлен. Возможные значения: 0 - не подготовлен; 1 - подготовлен. Поведение свойства: - *только для чтения*
refresh	string	Период автоматического обновления. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 1h). По умолчанию: 30s.
rows_per_page	integer	Количество строк объектов, отображаемых на странице. По умолчанию: 50.
surname	string	Фамилия пользователя.
theme	string	Тема пользователя. Возможные значения: default - *(по умолчанию)* системная тема по умолчанию; blue-theme - синяя; blue-classic-theme - синяя (классическая); dark-theme - темная; dark-blue-theme - темно-синяя; dark-classic-theme - темная (классическая); hc-light - светлая с высокой контрастностью; hc-dark - темная с высокой контрастностью.
ts_provisioned	timestamp	Время выполнения последней операции подготовки. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *суперадминистратор*
url	string	URL страницы, на которую перенаправляется пользователь после входа.

Property	Type	Description
userdirectoryid	ID	Идентификатор каталога пользователей, с которым связан пользователь. Используется для подготовки (создания или обновления), а также для входа пользователя, связанного с каталогом пользователей. При выполнении операций входа значение этого свойства имеет приоритет над свойством `userdirectoryid` групп пользователей, в которые входит пользователь. По умолчанию: 0. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *суперадминистратор*
timezone	string	Часовой пояс пользователя, например, `Europe/London`, `UTC`. По умолчанию: `default` - системный часовой пояс по умолчанию. Полный список поддерживаемых часовых поясов приведен в документации PHP.

Мультимедиа

Объект мультимедиа имеет следующие свойства.

Property	Type	Description
mediaid	ID	Идентификатор мультимедиа пользователя. Поведение свойства: - *только для чтения*
mediatypeid	ID	Идентификатор типа мультимедиа, используемого мультимедиа пользователя. Поведение свойства: - *обязательное*
sendto	string/array	Адрес, имя пользователя или другой идентификатор получателя. Если для `type` типа мультимедиа установлено значение ”Email” или ”Push”, значения представлены в виде массива. Для ”Push” массив может содержать либо подстановочный знак (`*`), либо один или несколько идентификаторов мобильных устройств. Если для `type` типа мультимедиа установлено значение ”Script”, ”SMS” или ”Webhook”, значение представлено в виде строки. Поведение свойства: - *обязательное*
active	integer	Указывает, включено ли мультимедиа. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.

Property	Type	Description
severity	integer	Уровни серьезности триггеров, о которых необходимо отправлять уведомления. Возможные значения битовой маски: 1 - Не классифицировано; 2 - Информация; 4 - Предупреждение; 8 - Средняя; 16 - Высокая; 32 - Катастрофа. Это поле является битовой маской; допускается любая сумма возможных значений битовой маски (например, 48 для уровней «Средняя», «Высокая» и «Катастрофа»). По умолчанию: 63.
period	string	Время, когда можно отправлять уведомления, в виде периода времени или пользовательских макросов, разделенных точкой с запятой. По умолчанию: 1-7,00:00-24:00.
provisioned	integer	Указывает, был ли пользователь подготовлен. Возможные значения: 0 - не подготовлен; 1 - подготовлен.
userdirectory_mediaid	ID	Идентификатор сопоставления мультимедиа каталога пользователей для подготовленного мультимедиа. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *Суперадминистратор*

user.checkAuthentication

Описание

`object user.checkAuthentication`

Этот метод проверяет и продлевает пользовательскую сессию.

> **Attention:**
> Вызов метода `user.checkAuthentication` с использованием параметра `sessionid` по умолчанию продлевает пользовательскую сессию.

Параметры

Метод принимает следующие параметры.

Parameter	Type	Description
extend	boolean	Следует ли продлевать сеанс пользователя. Значение по умолчанию: ”true”. Установка значения ”false” позволяет проверить сеанс пользователя без его продления. Поведение параметра: - *supported* если `sessionid` задан

Parameter	Type	Description
sessionid	string	Токен аутентификации пользователя. Поведение параметра: - *required* если `token` не задан
token	string	API-токен пользователя. Поведение параметра: - *required* если `sessionid` не задан

Возвращаемые значения

`(object)` Возвращает объект, содержащий информацию о пользователе.

В дополнение к стандартным свойствам пользователя возвращается следующая информация.

Property	Type	Description
auth_type	integer	Метод аутентификации по умолчанию для пользователя. См. свойство `authentication_type` объекта Authentication object для списка возможных значений.
debug_mode	integer	Включен или отключен режим отладки для пользователя. См. свойство `debug_mode` объекта User group object для списка возможных значений.
deprovisioned	boolean	Указывает, принадлежит ли пользователь к группе деактивированных пользователей.
gui_access	string	Метод аутентификации пользователя во веб-интерфейс. См. свойство `gui_access` объекта User group object для списка возможных значений.
secret	string	Случайная строка длиной 32 символа. Генерируется при входе пользователя в систему. Свойство `secret` не возвращается, если сеанс пользователя проверяется с использованием API-токена.
sessionid	string	Токен аутентификации, который необходимо использовать в следующих API-запросах. Свойство `sessionid` не возвращается, если сеанс пользователя проверяется с использованием API-токена.
type	integer	Тип пользователя. См. свойство `type` объекта Role object для списка возможных значений.
userip	string	IP-адрес пользователя.

Примеры

Проверка аутентификации с использованием токена аутентификации

Проверьте и продлите сеанс пользователя с помощью токена аутентификации пользователя, а также верните дополнительную информацию о пользователе.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
        "secret": "0e329b933e46984e49a5c1051ecd0751"
    },
    "id": 1
}
```


Проверка аутентификации с использованием API-токена

Проверяет сеанс пользователя с помощью API-токена пользователя и возвращает дополнительную информацию о пользователе.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
```

```
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "id": 1
}
```


Источник

CUser::checkAuthentication() в *ui/include/classes/api/services/CUser.php*.

user.create

Описание

`object user.create(object/array users)`

Этот метод позволяет создавать новых пользователей.

Note:
Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Note:
Надёжность пароля пользователя проверяется в соответствии с правилами политики паролей, определёнными API аутентификации. Дополнительную информацию см. в разделе объект authentication.

Параметры

`(object/array)` Пользователи для создания.

В дополнение к стандартным свойствам пользователя, метод принимает следующие параметры.

Parameter	Type	Description
usrgrps	array	Группы пользователей, в которые нужно добавить пользователя. Группы пользователей должны иметь определенным только свойство `usrgrpid`.
medias	array	Медиа пользователя, которые нужно создать.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных пользователей, указанных в свойстве `userids`. Порядок возвращаемых ID совпадает с порядком переданных пользователей.

Примеры

Создание пользователя

Создайте нового пользователя, добавьте его в группу пользователей и создайте для него новый способ оповещения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

См. также

- Аутентификация
- Медиа
- Группа пользователей
- Роль

Источник

CUser::create() в *ui/include/classes/api/services/CUser.php*.

user.delete

Описание

`object user.delete(array users)`

Этот метод позволяет удалять пользователей.

> **Note:**
> Этот метод доступен только для типа пользователя *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых пользователей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных пользователей, указанных в свойстве `userids`.

Примеры

Удаление нескольких пользователей

Удаление двух пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CUser::delete() в *ui/include/classes/api/services/CUser.php*.

user.get

Описание

`integer/array user.get(object parameters)`

Метод позволяет получать пользователей в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Note:
При запросе информации о пользовательских медиа, правах доступа или ролях пользователи типов *Admin* и *User* могут получать данные только о своем собственном пользователе. Пример см. в разделе Получение пользователей как *Admin*.

Параметр	Тип	Описание
mediaids	ID/array	Возвращать только пользователей, которые используют указанные медиа.
mediatypeids	ID/array	Возвращать только пользователей, которые используют указанные типы медиа.
userids	ID/array	Возвращать только пользователей с указанными ID.

Параметр	Тип	Описание
usrgrpids	ID/array	Возвращать только пользователей, которые принадлежат к указанным группам пользователей.
getAccess	flag	Добавляет дополнительную информацию о правах пользователя. Добавляет следующие свойства для каждого пользователя: `gui_access` - (`integer`) метод аутентификации пользователя во веб-интерфейсе. Список возможных значений см. в свойстве `gui_access` объекта группы пользователей. `debug_mode` - (`integer`) указывает, включен ли для пользователя режим отладки. Возможные значения: 0 - отладка отключена, 1 - отладка включена. `users_status` - (`integer`) указывает, отключен ли пользователь. Возможные значения: 0 - пользователь включен, 1 - пользователь отключен.
selectMedias	query	Возвращать медиа, используемые пользователем, в свойстве `medias`.
selectMediatypes	query	Возвращать типы медиа, используемые пользователем, в свойстве `mediatypes`. См. `mediatype.get` для ограничений в зависимости от типа пользователя.
selectUsrgrps	query	Возвращать группы пользователей, к которым принадлежит пользователь, в свойстве `usrgrps`. См. `usergroup.get` для ограничений в зависимости от типа пользователя.
selectRole	query	Возвращать роль пользователя в свойстве `role`.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сравнения. Не поддерживает свойства с типом данных `text`. Возможные свойства объекта пользователя для пользователей типов *Admin* и *User* при запросе данных о пользователях в их группе пользователей: `userid`, `name`, `surname`, `username`.
output	query	Свойства объекта пользователя, которые нужно вернуть. Пользователи типов *Admin* и *User* могут получать только следующие свойства: - Для своего собственного пользователя: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Для пользователей в своей группе пользователей: `userid`, `name`, `surname`, `username`. По умолчанию: `extend`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств, а значения — строки для поиска. Если дополнительные параметры не указаны, будет выполнен поиск `LIKE "%…%"`. Поддерживает только свойства с типами данных `string` и `text`. Возможные свойства объекта пользователя для пользователей типов *Admin* и *User* при запросе данных о пользователях в их группе пользователей: `name`, `surname`, `username`.

Параметр	Тип	Описание
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `userid`, `username`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение пользователей

Получите всех настроенных пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
```

```
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

Получение пользователей с ролью *Admin*

Пользователь типа *Admin* может получить подробные данные о собственной учетной записи и ограниченные данные о пользователях своей группы пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
```

```
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "3",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "0",
    "usrgrps": [
      {
        "usrgrpid": "7",
        "name": "Zabbix administrators",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "13",
        "name": "Internal",
        "gui_access": "1",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Super admin role",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
```

```
      "roleid": "4",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "1",
      "usrgrps": [
        {
          "usrgrpid": "8",
          "name": "Guests",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Disabled",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Получение данных пользователя

Получите данные пользователя с идентификатором «2».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```

См. также

- Медиа
- Тип медиа
- Группа пользователей
- Роль

Источник

CUser::get() в *ui/include/classes/api/services/CUser.php*.

user.login

Описание

`string/object user.login(object parameters)`

Этот метод позволяет вам войти в API и сгенерировать токен аутентификации.

> **Warning:**
> При использовании этого метода вам также необходимо выполнить user.logout, чтобы предотвратить создание большого количества открытых записей сеансов.

> **Attention:**
> Этот метод доступен только неаутентифицированным пользователям, которые не принадлежат ни к одной группе пользователей с включенной многофакторной аутентификацией.

Параметры

`(object)` Параметры, содержащие имя пользователя и пароль.

Метод принимает следующие параметры.

Parameter	Type	Description
password	string	Пароль пользователя. Parameter behavior: - *required*
username	string	Имя пользователя. Parameter behavior: - *required*
userData	flag	Возвращает информацию об аутентифицированном пользователе.

Возвращаемые значения

`(string/object)` Если используется параметр `userData`, возвращает объект с информацией об аутентифицированном пользователе.

Помимо стандартных свойств пользователя, возвращается следующая информация:

Property	Type	Description
auth_type	integer	Аутентификация по умолчанию для пользователя. См. свойство `authentication_type` объекта Authentication object для списка возможных значений.
debug_mode	integer	Включен или отключен режим отладки для пользователя. См. свойство `debug_mode` объекта User group object для списка возможных значений.
deprovisioned	boolean	Указывает, принадлежит ли пользователь к группе деактивированных пользователей.
gui_access	string	Способ аутентификации пользователя во веб-интерфейс. См. свойство `gui_access` объекта User group object для списка возможных значений.
mfaid	integer	ID метода MFA, который будет использоваться для пользователя при входе в систему. Возвращает ”0”, если MFA глобально отключена или отключена для всех групп пользователей, к которым принадлежит пользователь.
secret	string	Случайная строка из 32 символов. Генерируется при входе пользователя в систему.
sessionid	string	Токен аутентификации, который необходимо использовать в последующих API-запросах.
type	integer	Тип пользователя. См. свойство `type` объекта Role object для списка возможных значений.
userip	string	IP-адрес пользователя.

Note:
Если пользователь был успешно аутентифицирован после одной или нескольких неудачных попыток, метод вернет текущие значения свойств `attempt_clock`, `attempt_failed` и `attempt_ip`, а затем сбросит их.

Если параметр `userData` не используется, метод возвращает токен аутентификации, необходимый для аутентификации.

Примеры

Аутентификация пользователя

Аутентифицируйте пользователя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Запрос информации об аутентифицированном пользователе

Выполните аутентификацию и верните дополнительную информацию о пользователе.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "mfaid": "1",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "5b56eee8be445e98f0bd42b435736e42",
        "secret": "cd0ba923319741c6586f3d866423a8f4"
```

```
    },
    "id": 1
}
```

Смотрите также

- user.logout

Источник

CUser::login() в *ui/include/classes/api/services/CUser.php*.

user.logout

Описание

`string/object user.logout(array)`

Этот метод позволяет выйти из API и делает текущий токен аутентификации недействительным.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(  )` Метод принимает пустой массив.

Возвращаемые значения

`(  )` Возвращает `true`, если пользователь успешно выполнил выход.

Примеры

Выход из системы

Выйти из API.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Смотрите также

- user.login

Источник

CUser::login() в *ui/include/classes/api/services/CUser.php*.

user.provision

Описание

`object user.provision(object/array users)`

Этот метод позволяет выполнять подготовку пользователей LDAP.

Note:
Этот метод доступен только для типа пользователя *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы пользователей для подготовки.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы подготовленных пользователей в свойстве `userids`.

Примеры

Подготовка нескольких пользователей

Подготовьте двух пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Исходный код

CUser::provision() в *ui/include/classes/api/services/CUser.php*.

user.resettotp

Описание

`object user.resettotp(object/array users)`

Этот метод позволяет сбросить секреты TOTP пользователей.

Note:
Этот метод доступен только для пользователей типа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(array)` Идентификаторы пользователей, для которых нужно сбросить секреты TOTP.

Note:
Сеансы пользователей для указанных пользователей также будут удалены (за исключением пользователя, отправившего запрос).

Возвращаемые значения

(`object`) Возвращает объект, содержащий идентификаторы пользователей, для которых были сброшены секреты TOTP, в свойстве `userids`.

Примеры

Сброс секретов TOTP для нескольких пользователей

Сбросьте секреты TOTP для двух пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

См. также

- Объект MFA

Исходный код

CUser::resettotp() в *ui/include/classes/api/services/CUser.php*.

user.unblock

Описание

`object user.unblock(array userids)`

Этот метод позволяет разблокировать пользователей.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID пользователей для разблокировки.

Возвращаемые значения

( ) Возвращает объект, который содержит ID разблокированных пользователей, указанных в свойстве `userids`.

Примеры

Разблокировка нескольких пользователей

Разблокируйте двух пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CUser::unblock() в *ui/include/classes/api/services/CUser.php*.

user.update

Описание

`object user.update(object/array users)`

Этот метод позволяет обновлять существующих пользователей.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Note:
Надежность пароля пользователя проверяется в соответствии с правилами политики паролей, определенными Authentication API. Дополнительную информацию см. в разделе Authentication API.

Параметры

`(object/array)` Свойства пользователя, которые нужно обновить.

Свойство `userid` должно быть определено для каждого пользователя, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств пользователя, метод принимает следующие параметры.

Parameter	Type	Description
current_passwd	string	Текущий пароль пользователя. Значение этого параметра может быть пустой строкой, если пользователь связан с каталогом пользователей. Поведение параметра: - *только для записи* - *обязательно*, если `passwd` объекта User задан и пользователь изменяет собственный пароль

Parameter	Type	Description
usrgrps	array	Группы пользователей, которые заменят существующие группы пользователей. Группы пользователей должны иметь определенным только свойство `usrgrpid`.
medias	array	Медиа пользователя, которые заменят существующие непредоставленные медиа. Предоставленные медиа можно опустить при обновлении медиа.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных пользователей, указанных в свойстве`userids`.

Примеры

Переименование пользователя

Переименуйте пользователя в John Doe.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Изменение роли пользователя

Измените роль пользователя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
```

```
        ]
    },
    "id": 1
}
```

Смотрите также

- Аутентификация

Источник

CUser::update() в *ui/include/classes/api/services/CUser.php*.

Автоматическая регистрация

Этот класс предназначен для работы с автоматической регистрацией.

Ссылки на объекты:

- Автоматическая регистрация

Доступные методы:

- autoregistration.get - получить данные автоматической регистрации
- autoregistration.update - обновить данные автоматической регистрации

Объект авторегистрации

Следующие объекты напрямую связаны с API .

Автоматическая регистрация

Объект автоматической регистрации имеет следующие свойства.

Property	Type	Description
tls_accept	integer	Тип разрешенных входящих подключений для автоматической регистрации. Возможные значения: 1 - разрешить незашифрованные подключения; 2 - разрешить TLS с PSK; 3 - разрешить как незашифрованные подключения, так и подключения TLS с PSK.
tls_psk_identity	string	Идентификатор PSK; должен быть связан только с одним PSK (в рамках автоматической регистрации, узлов сети и прокси). Не включайте конфиденциальную информацию в идентификатор PSK, так как он передается по сети без шифрования, чтобы сообщить получателю, какой PSK использовать. Поведение свойства: - *только для записи*
tls_psk	string	Предварительно общий ключ (PSK); должен содержать не менее 32 шестнадцатеричных цифр. Поведение свойства: - *только для записи*

autoregistration.get

Описание

`object autoregistration.get(object parameters)`

Этот метод позволяет получить объект автрегистрирования в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типа пользователя *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает только один параметр.

Parameter	Type	Description
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

`(object)` Возвращает объект авторегистрации.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
    "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
    "tls_accept": "3"
    },
    "id": 1
}
```

Источник

CAutoregistration::get() в *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Описание

`object autoregistration.update(object autoregistration)`

Этот метод позволяет обновлять существующую автрорегистрацию.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Свойства авторегистрации для обновления.

Возвращаемые значения

`(boolean)` Возвращает логическое значение true в качестве результата при успешном обновлении.

Примеры

Запрос:

```
{
   "jsonrpc": "2.0",
   "method": "autoregistration.update",
   "params": {
      "tls_accept": "3",
      "tls_psk_identity": "PSK 001",
      "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
   },
   "auth": "038e1d7b1735c6a5436ee9eae095879e",
   "id": 1
}
```

Ответ:

```
{
   "jsonrpc": "2.0",
   "result": true,
   "id": 1
}
```

Источник

CAutoregistration::update() в *ui/include/classes/api/services/CAutoregistration.php*.

Аутентификация

Этот класс предназначен для работы с настройками аутентификации.

Ссылки на объекты:

- Authentication

Доступные методы:

- authentication.get - получить настройки аутентификации
- authentication.update - обновить настройки аутентификации

Authentication object

Следующие объекты напрямую связаны с `authentication` API.

Аутентификация

Объект аутентификации имеет следующие свойства.

Свойство	Тип	Описание
authentication_type	integer	Аутентификация по умолчанию. Возможные значения: 0 - *(по умолчанию)* внутренняя; 1 - LDAP.
http_auth_enabled	integer	Аутентификация HTTP. Возможные значения: 0 - *(по умолчанию)* отключена; 1 - включена. Поведение свойства: - *поддерживается*, если `$ZBX_FEATURE_FLAGS['http_auth_enabled']` имеет значение `true` в файле конфигурации веб-интерфейса.

Свойство	Тип	Описание
http_login_form	integer	Форма входа по умолчанию. Возможные значения: 0 - *(по умолчанию)* форма входа Zabbix; 1 - форма входа HTTP. Поведение свойства: - *поддерживается*, если `$ZBX_FEATURE_FLAGS['http_auth_enabled']` имеет значение `true` в файле конфигурации веб-интерфейса.
http_strip_domains	string	Имя домена для удаления. Поведение свойства: - *поддерживается*, если `$ZBX_FEATURE_FLAGS['http_auth_enabled']` имеет значение `true` в файле конфигурации веб-интерфейса.
http_case_sensitive	integer	Учет регистра при входе по HTTP. Возможные значения: 0 - выключено; 1 - *(по умолчанию)* включено. Поведение свойства: - *поддерживается*, если `$ZBX_FEATURE_FLAGS['http_auth_enabled']` имеет значение `true` в файле конфигурации веб-интерфейса.
ldap_auth_enabled	integer	Аутентификация LDAP. Возможные значения: 0 - *(по умолчанию)* отключена; 1 - включена.
ldap_case_sensitive	integer	Учет регистра при входе через LDAP. Возможные значения: 0 - выключено; 1 - *(по умолчанию)* включено.
ldap_userdirectoryid	ID	Идентификатор каталога пользователей по умолчанию для аутентификации LDAP. Используется для групп пользователей, у которых `gui_access` имеет значение LDAP или «Системное значение по умолчанию». Поведение свойства: - *обязательно*, если `ldap_auth_enabled` имеет значение «Включена»
saml_auth_enabled	integer	Аутентификация SAML. Возможные значения: 0 - *(по умолчанию)* отключена; 1 - включена.
saml_case_sensitive	integer	Учет регистра при входе через SAML. Возможные значения: 0 - выключено; 1 - *(по умолчанию)* включено.
passwd_min_length	integer	Минимальная длина пароля. Допустимый диапазон значений: от 1 до 70. По умолчанию: 8.

Свойство	Тип	Описание
passwd_check_rules	integer	Правила проверки пароля. Возможные значения битовой маски: 0 - проверять длину пароля; 1 - проверять использование в пароле прописных и строчных латинских букв; 2 - проверять использование в пароле цифр; 4 - проверять использование в пароле специальных символов; 8 - *(по умолчанию)* проверять, что пароль отсутствует в списке часто используемых паролей и не содержит производных от слова «Zabbix», имени, фамилии или имени пользователя. Это поле является битовой маской; допускается любая сумма возможных значений (например, 15 для проверки всех правил).
ldap_jit_status	integer	Состояние автоматического создания пользователей LDAP. Возможные значения: 0 - отключено для настроенных IdP LDAP; 1 - включено для настроенных IdP LDAP.
saml_jit_status	integer	Состояние автоматического создания пользователей SAML. Возможные значения: 0 - отключено для настроенных IdP SAML; 1 - включено для настроенных IdP SAML.
jit_provision_interval	string	Интервал времени между запросами автоматического создания пользователей для вошедшего пользователя. Принимает количество секунд (например, 3600) или значение с суффиксом времени (например, 3600s, 60m, 1h, 1d, 1M, 1y), включая поддержку месяцев и лет. Минимальное значение: 1h. По умолчанию: 1h. Доступно только для автоматического создания пользователей LDAP.
disabled_usrgrpid	ID	Идентификатор группы пользователей, в которую следует назначить пользователя после отмены его автоматического создания. Группа пользователей должна быть отключена; после настройки ее нельзя включить или удалить. Поведение свойства: - *обязательно*, если `ldap_jit_status` имеет значение «Включено для настроенных IdP LDAP» или `saml_jit_status` имеет значение «Включено для настроенных IdP SAML»
mfa_status	integer	Многофакторная аутентификация. Возможные значения: 0 - отключена (для всех настроенных методов MFA); 1 - включена (для всех настроенных методов MFA).
mfaid	ID	Метод MFA по умолчанию для групп пользователей с включенной MFA. Поведение свойства: - *обязательно*, если `mfa_status` имеет значение «Включена»

authentication.get

Описание

`object authentication.get(object parameters)`

Метод позволяет получить объект аутентификации в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типа пользователя *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает только один параметр.

Parameter	Type	Description
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

`(object)` Возвращает объект аутентификации.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "authentication_type": "0",
        "http_auth_enabled": "0",
        "http_login_form": "0",
        "http_strip_domains": "",
        "http_case_sensitive": "1",
        "ldap_auth_enabled": "0",
        "ldap_case_sensitive": "1",
        "ldap_userdirectoryid": "0",
        "saml_auth_enabled": "0",
        "saml_case_sensitive": "0",
        "passwd_min_length": "8",
        "passwd_check_rules": "15",
        "jit_provision_interval": "1h",
        "saml_jit_status": "0",
        "ldap_jit_status": "0",
        "disabled_usrgrpid": "9",
        "mfa_status": "0",
        "mfaid": "0"
    },
    "id": 1
}
```

Источник

CAuthentication::get() в *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Описание

`object authentication.update(object authentication)`

Этот метод позволяет обновлять существующие настройки аутентификации.

> **Note:**
> Этот метод доступен только для типа пользователя *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Свойства аутентификации подлежат обновлению.

Возвращаемые значения

`(array)` Возвращает массив с именами обновленных параметров.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Источник

CAuthentication::update() в *ui/include/classes/api/services/CAuthentication.php*.

Веб-сценарий

Этот класс предназначен для работы с веб-сценариями.

Ссылки на объекты:

- Веб-сценарий
- Шаг сценария
  - Поле HTTP
- Тег веб-сценария

Доступные методы:

- httptest.create - создать новые веб-сценарии
- httptest.delete - удалить веб-сценарии
- httptest.get - получить веб-сценарии
- httptest.update - обновить веб-сценарии

Объект веб-сценария

Следующие объекты напрямую связаны с API `webcheck`.

Веб-сценарий

Объект веб-сценария имеет следующие свойства.

Свойство	Тип	Описание
httptestid	ID	Идентификатор веб-сценария. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
hostid	ID	Идентификатор узла сети, которому принадлежит веб-сценарий. Поведение свойства: - *константа* - *обязательно* для операций создания
name	string	Имя веб-сценария. Поведение свойства: - *обязательно* для операций создания
agent	string	Строка агента пользователя, которая будет использоваться веб-сценарием. По умолчанию: Zabbix
authentication	integer	Метод аутентификации, который будет использоваться веб-сценарием. Возможные значения: 0 - *(по умолчанию)* отсутствует; 1 - базовая HTTP-аутентификация; 2 - аутентификация NTLM; 3 - аутентификация Kerberos; 4 - дайджест-аутентификация.
delay	string	Интервал выполнения веб-сценария. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательский макрос. По умолчанию: 1m.
headers	array	HTTP-заголовки, которые будут отправляться при выполнении запроса.
http_password	string	Пароль, используемый для базовой HTTP-аутентификации, аутентификации NTLM, Kerberos или дайджест-аутентификации.
http_proxy	string	Прокси, который будет использоваться веб-сценарием, в формате *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Имя пользователя, используемое для базовой HTTP-аутентификации, аутентификации NTLM, Kerberos или дайджест-аутентификации.
retries	integer	Количество попыток веб-сценария выполнить каждый шаг перед сбоем. По умолчанию: 1.
ssl_cert_file	string	Имя файла сертификата SSL, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_file	string	Имя файла закрытого ключа SSL, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль закрытого ключа SSL.

Свойство	Тип	Описание
status	integer	Определяет, включен ли веб-сценарий. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
templateid	ID	Идентификатор родительского веб-сценария шаблона. Поведение свойства: - *только для чтения*
variables	array	Переменные веб-сценария.
verify_host	integer	Определяет, проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять узел сети; 1 - проверять узел сети.
verify_peer	integer	Определяет, проверять ли подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять узел; 1 - проверять узел.
uuid	string	Глобальный уникальный идентификатор, используемый для связывания импортированных веб-сценариев с уже существующими. Создается автоматически, если не указан. Поведение свойства: - *поддерживается*, если веб-сценарий принадлежит шаблону

Шаг сценария

Объект шага сценария определяет конкретную проверку веб-сценария. Он имеет следующие свойства.

Property	Type	Description
name	string	Имя шага сценария. Поведение свойства: - *обязательное*
no	integer	Порядковый номер шага в веб-сценарии. Поведение свойства: - *обязательное*
url	string	URL-адрес для проверки. Поведение свойства: - *обязательное*
follow_redirects	integer	Следует ли переходить по HTTP-перенаправлениям. Возможные значения: 0 - не переходить по перенаправлениям; 1 - *(по умолчанию)* переходить по перенаправлениям.
headers	array	HTTP-заголовки, отправляемые при выполнении запроса. Заголовки шага сценария заменяют заголовки, указанные для веб-сценария.
posts	string/array	Переменные HTTP POST в виде строки (необработанные данные POST) или массива полей HTTP (данные полей формы).
required	string	Текст, который должен присутствовать в ответе.

Property	Type	Description
retrieve_mode	integer	Часть HTTP-ответа, которую должен получить шаг сценария. Возможные значения: 0 - *(по умолчанию)* только тело; 1 - только заголовки; 2 - заголовки и тело.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделенные запятыми.
timeout	string	Тайм-аут запроса в секундах. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 1h) или пользовательский макрос. По умолчанию: 15s. Максимум: 1h. Минимум: 1s.
variables	array	Переменные шага сценария.
query_fields	array	Поля запроса - массив полей HTTP, добавляемых в URL при выполнении запроса.

Объект HTTP-поля

Объект HTTP-поля определяет имя и значение, которые используются для задания переменных веб-сценария, HTTP-заголовков, а также полей POST или полей запроса. Он имеет следующие свойства.

Property	Type	Description
name	string	Имя заголовка/переменной/поля POST или GET. Property behavior: - *required*
value	string	Значение заголовка/переменной/поля POST или GET. Property behavior: - *required*

Тег сценария веб-проверки

Объект тега сценария веб-проверки имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега сценария веб-проверки. Property behavior: - *required*
value	string	Значение тега сценария веб-проверки. Property behavior: - *read-only*.
object	integer	Тип объекта, от которого был унаследован тег. Possible values: 0 - template; 1 - host. Property behavior: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Property behavior: - *read-only*.

httptest.create

Описание

`object httptest.create(object/array webScenarios)`

Этот метод позволяет создавать новые веб-сценарии.

> **Note:**
> При создании веб-сценария автоматически будет создан набор элементов данных веб-мониторинга.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object/array)` Веб-сценарии для создания.

В дополнение к стандартным свойствам веб-сценария метод принимает следующие параметры.

Parameter	Type	Description
steps	array	Шаги сценария. Поведение параметра: - *required*
tags	array	Теги веб-сценария.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных веб-сценариев под свойством `httptestids`. Порядок возвращаемых ID совпадает с порядком переданных веб-сценариев.

Примеры

Создание веб-сценария

Создайте веб-сценарий для мониторинга домашней страницы компании. Сценарий будет состоять из двух шагов: проверки домашней страницы и страницы "About", чтобы убедиться, что они возвращают код состояния HTTP 200.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.create",
    "params": {
        "name": "Homepage check",
        "hostid": "10085",
        "steps": [
            {
                "name": "Homepage",
                "url": "http://example.com",
                "status_codes": "200",
                "no": 1
            },
            {
                "name": "Homepage / About",
                "url": "http://example.com/about",
                "status_codes": "200",
                "no": 2
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

См. также

- Шаг сценария

Источник

CHttpTest::create() в *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Описание

`object httptest.delete(array webScenarioIds)`

Этот метод позволяет удалять веб-сценарии.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` ID удаляемых веб-сценариев.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных веб-сценариев в свойстве `httptestids`.

Примеры

Удаление нескольких веб-сценариев

Удаление двух веб-сценариев.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Источник

CHttpTest::delete() в *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Описание

`integer/array httptest.get(object parameters)`

Метод позволяет получать веб-сценарии в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
groupids	ID/array	Возвращать только веб-сценарии, которые принадлежат указанным группам узлов сети.
hostids	ID/array	Возвращать только веб-сценарии, которые принадлежат указанным узлам сети.
httptestids	ID/array	Возвращать только веб-сценарии с указанными ID.
inherited	boolean	Если установлено значение `true`, возвращать только веб-сценарии, унаследованные от шаблона.
inheritedTags	boolean	Возвращать веб-сценарии, у которых указанные `tags` также присутствуют в шаблоне/узле сети/связанных шаблонах. Возможные значения: `true` - шаблон/узел сети/связанные шаблоны также должны иметь указанные теги; `false` - *(по умолчанию)* теги из шаблона/узла сети/связанных шаблонов игнорируются.
monitored	boolean	Если установлено значение `true`, возвращать только включенные веб-сценарии, которые принадлежат контролируемым узлам сети.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые находятся в шаблоне/узле сети/связанных шаблонах.
templated	boolean	Если установлено значение `true`, возвращать только веб-сценарии, которые принадлежат шаблонам.
templateids	ID/array	Возвращать только веб-сценарии, которые принадлежат указанным шаблонам.
expandName	flag	Расширять макросы в имени веб-сценария.
expandStepName	flag	Расширять макросы в именах шагов сценария.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.

Parameter	Type	Description
tags	array	Возвращать только веб-сценарии с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все веб-сценарии. Возможные значения оператора: 0 - *(по умолчанию)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
selectHosts	query	Возвращать узлы сети, к которым принадлежит веб-сценарий, в виде массива в свойстве `hosts`.
selectSteps	query	Возвращать шаги веб-сценария в свойстве `steps`. Поддерживает `count`.
selectTags	query	Возвращать теги веб-сценария в свойстве `tags`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `httptestid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использован параметр `countOutput`.

Примеры

Получение веб-сценария

Получить все данные о веб-сценарии "9".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "uuid": "",
            "headers": [],
            "variables": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
                    "no": "2",
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [],
                    "query_fields": []
                }
            ]
```

```
        }
    ],
    "id": 1
}
```

См. также

- Узел сети
- Шаг сценария

Источник

CHttpTest::get() в *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Описание

`object httptest.update(object/array webScenarios)`

Этот метод позволяет обновлять существующие веб-сценарии.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства веб-сценария, которые необходимо обновить.

Для каждого веб-сценария должно быть определено свойство `httptestid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств веб-сценария, метод принимает следующие параметры.

Parameter	Type	Description
steps	array	Шаги сценария для замены существующих шагов.
tags	array	Теги веб-сценария.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных веб-сценариев, указанных в свойстве `httptestid`.

Примеры

Включение веб-сценария

Включите веб-сценарий, то есть установите для него статус "0".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

См. также

- Шаг сценария

Источник

CHttpTest::update() в *ui/include/classes/api/services/CHttpTest.php*.

Граф

Этот класс предназначен для работы с графиками.

Ссылки на объекты:

- Graph

Доступные методы:

- graph.create - создать новые графики
- graph.delete - удалить графики
- graph.get - получить графики
- graph.update - обновить графики

Объект graph

Следующие объекты напрямую связаны с API `graph`.

График

Объект графика имеет следующие свойства.

Property	Type	Description
graphid	ID	ID графика. Property behavior: - *read-only* - *required* for update operations
height	integer	Высота графика в пикселях. Property behavior: - *required* for create operations
name	string	Имя графика. Property behavior: - *required* for create operations
width	integer	Ширина графика в пикселях. Property behavior: - *required* for create operations
flags	integer	Origin графика. Possible values: 0 - *(default)* a plain graph; 4 - a graph converted from prototype. Property behavior: - *read-only*

Property	Type	Description
graphtype	integer	Тип макета графика. Possible values: 0 - *(default)* normal; 1 - stacked; 2 - pie; 3 - exploded.
percent_left	float	Левый процентиль. Default: 0.
percent_right	float	Правый процентиль. Default: 0.
show_3d	integer	Показывать ли круговые и развернутые графики в 3D. Possible values: 0 - *(default)* show in 2D; 1 - show in 3D.
show_legend	integer	Показывать ли легенду на графике. Possible values: 0 - hide; 1 - *(default)* show.
show_work_period	integer	Показывать ли рабочее время на графике. Possible values: 0 - hide; 1 - *(default)* show.
show_triggers	integer	Показывать ли линию триггера на графике. Possible values: 0 - hide; 1 - *(default)* show.
templateid	ID	ID родительского шаблона графика. Property behavior: - *read-only*
yaxismax	float	Фиксированное максимальное значение для оси Y. Default: 100.
yaxismin	float	Фиксированное минимальное значение для оси Y. Default: 0.
ymax_itemid	ID	ID элемента данных, который используется как максимальное значение для оси Y. If a user has no access to the specified item, the graph is rendered as if `ymax_type` is set to ”calculated”.
ymax_type	integer	Метод расчета максимального значения для оси Y. Possible values: 0 - *(default)* calculated; 1 - fixed; 2 - item.
ymin_itemid	ID	ID элемента данных, который используется как минимальное значение для оси Y. If a user has no access to the specified item, the graph is rendered as if `ymin_type` is set to ”calculated”.

Property	Type	Description
ymin_type	integer	Метод расчета минимального значения для оси Y. Possible values: 0 - *(default)* calculated; 1 - fixed; 2 - item.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных графиков с уже существующими. Если не указан, генерируется автоматически. Property behavior: - *supported* if the graph belongs to a template

graph.create

Описание

`object graph.create(object/array graphs)`

Этот метод позволяет создавать новые графики.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Графики для создания.

В дополнение к стандартным свойствам графика, метод принимает следующие параметры.

Parameter	Type	Description
gitems	array	Элементы графика, которые будут созданы для графика. Поведение параметра: - *required*

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных графиков, указанных в свойстве `graphids`. Порядок возвращаемых ID совпадает с порядком переданных графиков.

Примеры

Создание графика

Создайте график с двумя элементами данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graph.create",
    "params": {
        "name": "MySQL bandwidth",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
```

```
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

См. также

- Элемент данных графика

Источник

CGraph::create() в *ui/include/classes/api/services/CGraph.php*.

graph.delete

Описание

`object graph.delete(array graphIds)`

Этот метод позволяет удалять графики.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых графиков.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных графиков, указанных в свойстве `graphids`.

Примеры

Удаление нескольких графиков

Удаление двух графиков.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Источник

CGraph::delete() в *ui/include/classes/api/services/CGraph.php*.

graph.get

Описание

`integer/array graph.get(object parameters)`

Метод позволяет получать графики в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
graphids	ID/array	Возвращать только графики с указанными ID.
groupids	ID/array	Возвращать только графики, которые принадлежат узлам сети или шаблонам в указанных группах узлов сети или группах шаблонов.
templateids	ID/array	Возвращать только графики, которые принадлежат указанным шаблонам.
hostids	ID/array	Возвращать только графики, которые принадлежат указанным узлам сети.
itemids	ID/array	Возвращать только графики, которые содержат указанные элементы данных.
templated	boolean	Если установлено в `true`, возвращать только графики, которые принадлежат шаблонам.
inherited	boolean	Если установлено в `true`, возвращать только графики, унаследованные от шаблона.
expandName	flag	Разворачивать макросы в имени графика.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения графика. Объект обнаружения графика связывает обнаруженный график с прототипом графика, из которого он был обнаружен. Он имеет следующие свойства: `parent_graphid` - `(ID)` ID прототипа графика, из которого был создан график; `status` - `(int)` статус обнаружения графика: 0 - *(по умолчанию)* график обнаружен, 1 - график больше не обнаружен; `ts_delete` - `(timestamp)` время, когда будет удален график, который больше не обнаружен.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом низкоуровневого обнаружения, которое создало график.

Parameter	Type	Description
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, к которым принадлежит график.
selectTemplateGroups	query	Возвращать свойство `templategroups` с группами шаблонов, к которым принадлежит график.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, к которым принадлежит график.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, к которым принадлежит график.
selectItems	query	Возвращать свойство `items` с элементами данных, используемыми в графике.
selectGraphItems	query	Возвращать свойство `gitems` с элементами данных, используемыми в графике.
filter	object	Возвращать только те результаты, которые в точности соответствуют указанному фильтру. Принимает объект, где ключи - это имена свойств, а значения - либо одно значение, либо массив значений, с которыми выполняется сравнение. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит график; `hostid` - ID узла сети, к которому принадлежит график.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `graphid`, `name`, `graphtype`.
countOutput	boolean	Эти параметры описаны в комментарии к справочнику.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGraphDiscovery	query	Возвращать свойство `graphDiscovery` с объектом обнаружения графика. Объекты обнаружения графика связывают график с прототипом графика, из которого он был создан. Этот запрос **устарел**, вместо него используйте `selectDiscoveryData`.

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение графиков с узлов сети

Получить все графики с узла сети "10107" и отсортировать их по имени.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
```

```
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "612",
            "name": "CPU jumps",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "439",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "613",
            "name": "CPU load",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "433",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "614",
            "name": "CPU utilization",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "387",
```

```
            "show_work_period": "1",
            "show_triggers": "0",
            "graphtype": "1",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "1",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "645",
            "name": "Disk space usage /",
            "width": "600",
            "height": "340",
            "yaxismin": "0",
            "yaxismax": "0",
            "templateid": "0",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "4"
        }
    ],
    "id": 1
}
```


См. также

- Правило обнаружения
- Элемент графика
- Элемент данных
- узел сети
- Группа узлов сети
- шаблон
- Группа шаблонов

Источник

CGraph::get() в *ui/include/classes/api/services/CGraph.php*.

graph.update

Описание

`object graph.update(object/array graphs)`

Этот метод позволяет обновлять существующие графики.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

(object/array) Свойства графика, которые необходимо обновить.

Для каждого графика должно быть определено свойство graphid, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам графика метод принимает следующие параметры.

Parameter	Type	Description
gitems	array	Элементы графика для замены существующих элементов графика. Если у элемента графика определено свойство gitemid, он будет обновлен, в противном случае будет создан новый элемент графика.

Возвращаемые значения

(object) Возвращает объект, который содержит ID обновленных графиков, указанных в свойстве graphids.

Примеры

Установка максимума для шкалы Y

Установите максимум шкалы Y в фиксированное значение 100.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Источник

CGraph::update() в *ui/include/classes/api/services/CGraph.php*.

Группа пользователей

Этот класс предназначен для работы с группами пользователей.

Ссылки на объекты:

- Группа пользователей
- Разрешение
- Разрешение на основе тегов

Доступные методы:

- usergroup.create - создать новые группы пользователей
- usergroup.delete - удалить группы пользователей
- usergroup.get - получить группы пользователей
- usergroup.update - обновить группы пользователей

Объект группы пользователей

Следующие объекты напрямую связаны с API `usergroup`.

Группа пользователей

Объект группы пользователей имеет следующие свойства.

Property	Type	Description
usrgrpid	ID	Идентификатор группы пользователей. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Название группы пользователей. Поведение свойства: - *обязательно* для операций создания
debug_mode	integer	Включен или отключен режим отладки. Возможные значения: 0 - *(по умолчанию)* отключен; 1 - включен.
gui_access	integer	Метод аутентификации в веб-интерфейсе пользователей группы. Возможные значения: 0 - *(по умолчанию)* использовать системный метод аутентификации по умолчанию; 1 - использовать внутреннюю аутентификацию; 2 - использовать аутентификацию LDAP; 3 - запретить доступ к веб-интерфейсу.
mfa_status	integer	Включена или отключена MFA для пользователей группы. Возможные значения: 0 - отключена (для всех настроенных методов MFA); 1 - включена (для всех настроенных методов MFA).
mfaid	ID	Метод MFA, используемый для пользователей группы. Поведение свойства: - *поддерживается* для пользователей типа *Super admin* - *поддерживается*, если для свойства `mfa_status` объекта аутентификации установлено значение «Enabled»
proxy_mode	integer	Режим обработки доступа к прокси, не включенным в группы прокси. Возможные значения: 0 - список запрещенных; 1 - *(по умолчанию)* список разрешенных. Поведение свойства: - *поддерживается* для пользователей типа *Super admin*.
proxy_group_mode	integer	Режим обработки доступа к группам прокси (и их прокси). Возможные значения: 0 - список запрещенных; 1 - *(по умолчанию)* список разрешенных. Поведение свойства: - *поддерживается* для пользователей типа *Super admin*.

Property	Type	Description
users_status	integer	Включена или отключена группа пользователей. Для депровиженированных пользователей группу пользователей нельзя включить. Возможные значения: 0 - *(по умолчанию)* включена; 1 - отключена.
userdirectoryid	ID	Идентификатор каталога пользователей, используемого для аутентификации. Поведение свойства: - *поддерживается* для пользователей типа *Super admin* - *поддерживается*, если для `gui_access` установлено значение «использовать системный метод аутентификации по умолчанию» или «использовать аутентификацию LDAP»

Разрешение

Объект разрешения имеет следующие свойства.

Property	Type	Description
id	ID	ID группы узлов сети или группы шаблонов, для которой нужно добавить разрешение. Поведение свойства: - *required*
permission	integer	Уровень доступа к группе узлов сети или группе шаблонов. Возможные значения: 0 - доступ запрещен; 2 - доступ только для чтения; 3 - доступ для чтения и записи. Поведение свойства: - *required* для операций создания

Разрешение на основе тегов

Объект разрешения на основе тегов имеет следующие свойства.

Property	Type	Description
groupid	ID	ID группы узлов сети, для которой нужно добавить разрешение. Property behavior: - *required*
tag	string	Имя тега.
value	string	Значение тега.

APM-based permission

The APM-based permission object has the following properties.

Property	Type	Description
key	string	APM resource attribute key. Property behavior: - *required*
value	string	APM resource attribute value.

Property	Type	Description
signal_type	integer	APM resource attribute category. Possible values: -1 - *(default)* All; 0 - Traces; 1 - Metrics; 2 - Logs.

usergroup.create

Описание

`object usergroup.create(object/array userGroups)`

Этот метод позволяет создавать новые группы пользователей.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Группы пользователей для создания.

Помимо стандартных свойств группы пользователей, метод принимает следующие параметры.

Parameter	Type	Description
hostgroup_rights	object/array	Разрешения для групп узлов сети, назначаемые группе пользователей.
proxies	object/array	Прокси, которые будут связаны с группой пользователей. Прокси должны иметь только свойство `proxyid`.
proxy_groups	object/array	Группы прокси, которые будут связаны с группой пользователей. Группы прокси должны иметь только свойство `proxy_groupid`.
templategroup_rights	object/array	Разрешения для групп шаблонов, назначаемые группе пользователей.
tag_filters	array	Разрешения на основе тегов, назначаемые группе пользователей.
users	object/array	Пользователи, добавляемые в группу пользователей. Пользователь должен иметь только свойство `userid`.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных групп пользователей в свойстве `usrgrpids`. Порядок возвращаемых идентификаторов соответствует порядку переданных групп пользователей.

Примеры

Создание группы пользователей

Создайте группу пользователей *Operation managers* с запрещенным доступом к группе узлов сети "2" и добавьте в нее пользователя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
```

```
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Смотрите также

- Права доступа

Исходный код

CUserGroup::create() в *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Описание

`object usergroup.delete(array userGroupIds)`

Этот метод позволяет удалять группы пользователей.

Attention:
Группа пользователей Deprovisioned (группа пользователей, указанная для `disabled_usrgrpid` в Authentication) не может быть удалена.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе User roles.

Параметры

( ) ID удаляемых групп пользователей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удалённых групп пользователей, указанных в свойстве `usrgrpids`.

Примеры

Удаление нескольких групп пользователей

Удаление двух групп пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Источник

CUserGroup::delete() в *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Описание

`integer/array usergroup.get(object parameters)`

Метод позволяет получить группы пользователей в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
mfaids	ID/array	Возвращать только группы пользователей с указанными методами MFA. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*
mfa_status	integer	Возвращать только группы пользователей с указанным статусом MFA. Список поддерживаемых статусов приведен на странице групп пользователей.
proxyids	ID/array	Возвращать только группы пользователей, содержащие указанные прокси. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*.
proxy_groupids	ID/array	Возвращать только группы пользователей, содержащие указанные группы прокси. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*.
selectProxies	ID/array	Возвращать свойство `proxies` с прокси группы пользователей. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*.

Параметр	Тип	Описание
selectProxyGroups	ID/array	Возвращать свойство `proxy_groups` с группами прокси группы пользователей. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*.
status	integer	Возвращать только группы пользователей с указанным статусом. Список поддерживаемых статусов приведен на странице групп пользователей.
userids	ID/array	Возвращать только группы пользователей, содержащие указанных пользователей.
usrgrpids	ID/array	Возвращать только группы пользователей с указанными идентификаторами.
selectTagFilters	query	Возвращать разрешения групп пользователей на основе тегов в свойстве `tag_filters`.
selectUsers	query	Возвращать пользователей из группы пользователей в свойстве `users`. Ограничения в зависимости от типа пользователя см. в описании метода `user.get`.
selectHostGroupRights	query	Возвращать разрешения группы пользователей для групп узлов сети в свойстве `hostgroup_rights`. Список уровней доступа к группам узлов сети приведен на странице групп пользователей.
selectTemplateGroupRights	query	Возвращать разрешения группы пользователей для групп шаблонов в свойстве `templategroup_rights`. Список уровней доступа к группам шаблонов приведен на странице групп пользователей.
limitSelects	integer	Ограничивает количество записей, возвращаемых вложенными выборками.
output	query	Возвращаемые свойства объекта группы пользователей. Пользователи типов *Admin* и *User* могут получать только следующие свойства: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. По умолчанию: `extend`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `usrgrpid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном разделе.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение включенных групп пользователей

Получить все включенные группы пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
```

```
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


См. также

- User

Источник

CUserGroup::get() в *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Описание

`object usergroup.update(object/array userGroups)`

Этот метод позволяет обновлять существующие группы пользователей.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства групп пользователей, которые необходимо обновить.

Свойство `usrgrpid` должно быть определено для каждой группы пользователей, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств группы пользователей, метод принимает следующие параметры.

Parameter	Type	Description
hostgroup_rights	object/array	Права доступа к группам узлов сети, которыми необходимо заменить текущие права доступа, назначенные группе пользователей.
proxies	object/array	Прокси, которые необходимо связать с группой пользователей. Прокси должны иметь только свойство `proxyid`.
proxy_groups	object/array	Группы прокси, которые необходимо связать с группой пользователей. Группы прокси должны иметь только свойство `proxy_groupid`.
templategroup_rights	object/array	Права доступа к группам шаблонов, которыми необходимо заменить текущие права доступа, назначенные группе пользователей.
tag_filters	array	Права доступа на основе тегов, которыми необходимо заменить текущие права доступа, назначенные группе пользователей.

Parameter	Type	Description
users	object/array	Пользователи, которыми необходимо заменить текущих пользователей, назначенных группе пользователей. Пользователь должен иметь только свойство `userid`.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных групп пользователей, указанных в свойстве `usrgrpids`.

Примеры

Включение группы пользователей и обновление прав доступа

Включите группу пользователей и предоставьте ей доступ на чтение и запись к группам узлов сети ”2” и ”4”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Смотрите также

- Права доступа

Источник

CUserGroup::update() в *ui/include/classes/api/services/CUserGroup.php*.

Группа прокси

Этот класс предназначен для работы с группами прокси.

Ссылки на объекты:

- Группа прокси

Доступные методы:

- proxygroup.create - создать новые группы прокси

- proxygroup.delete - удалить группы прокси
- proxygroup.get - получить группы прокси
- proxygroup.update - обновить группы прокси

Объект группы прокси

Следующие объекты напрямую связаны с API `proxygroup`.

Группа прокси

Объект группы прокси имеет следующие свойства.

Property	Type	Description
proxy_groupid	ID	Идентификатор группы прокси. Поведение свойства: - *только для чтения* - *обязательное* для операций обновления
name	string	Название группы прокси. Поведение свойства: - *обязательное* для операций создания
description	text	Описание группы прокси.
failover_delay	string	Период, в течение которого прокси в группе прокси должен обмениваться данными с сервером Zabbix, чтобы считаться доступным. Поддерживаются суффиксы времени (например, 30s, 1m). Поддерживаются пользовательские макросы. Возможные значения: 10s-15m. Значение по умолчанию: 1m.
min_online	string	Минимальное количество доступных прокси, необходимое для сохранения доступности группы прокси. Поддерживаются пользовательские макросы. Диапазон возможных значений: 1-1000. Значение по умолчанию: 1.
state	integer	Состояние группы прокси. Возможные значения: 0 - Неизвестно; 1 - Недоступна; 2 - Восстанавливается; 3 - Доступна; 4 - Деградирует. Поведение свойства: - *только для чтения*

proxygroup.create

Описание

`object proxygroup.create(object/array proxyGroups)`

Этот метод позволяет создавать новые группы прокси.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Группы прокси для создания.

Метод принимает группы прокси со стандартными свойствами группы прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных групп прокси в свойстве `proxy_groupids`. Порядок возвращаемых идентификаторов соответствует порядку переданных групп прокси.

Примеры

Создание группы прокси

Создайте группу прокси с пользовательскими настройками.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CProxyGroup::create() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Описание

`object proxygroup.delete(array proxyGroupIds)`

Этот метод позволяет удалять группы прокси.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(array)` Идентификаторы групп прокси для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных групп прокси в свойстве `proxy_groupids`.

Примеры

Удаление нескольких групп прокси

Удаление двух групп прокси.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

Источник

CProxyGroup::delete() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Описание

`integer/array proxygroup.get(object parameters)`

Метод позволяет получить группы прокси в соответствии с указанными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей. Возвращаются только группы прокси, разрешенные параметром *Proxy access list* в группах пользователей пользователя.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
proxy_groupids	ID/array	Возвращать только группы прокси с указанными ID.
proxyids	ID/array	Возвращать только группы прокси, которые содержат указанные прокси.
selectProxies	query	Возвращать свойство `proxies` с прокси, которые принадлежат группе прокси. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `proxy_groupid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	

Parameter	Type	Description
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение всех групп прокси

Получение всех настроенных групп прокси с прокси.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxy_groupid": "1",
            "name": "         1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "         2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
```

```
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Прокси

Источник

CProxyGroup::get() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Описание

`object proxygroup.update(object/array proxyGroups)`

Этот метод позволяет обновлять существующие группы прокси.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства группы прокси, которые необходимо обновить.

Свойство `proxy_groupid` должно быть определено для каждой группы прокси, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Метод принимает группы прокси со стандартными свойствами группы прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных групп прокси в свойстве `proxy_groupids`.

Примеры

Изменение минимального числа онлайн-прокси

Измените минимальное число онлайн-прокси, необходимое для того, чтобы группа считалась онлайн.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Исходный код

CProxyGroup::update() в *ui/include/classes/api/services/CProxyGroup.php*.

Группа узлов сети

Этот класс предназначен для работы с группами узлов сети.

Ссылки на объекты:

- Группа узлов сети

Доступные методы:

- hostgroup.create - создать новые группы узлов сети
- hostgroup.delete - удалить группы узлов сети
- hostgroup.get - получить группы узлов сети
- hostgroup.massadd - добавить связанные объекты в группы узлов сети
- hostgroup.massremove - удалить связанные объекты из групп узлов сети
- hostgroup.propagate - распространить права доступа и фильтры тегов на подгруппы групп узлов сети
- hostgroup.update - обновить группы узлов сети

Объект группы узлов сети

Следующие объекты напрямую связаны с API `hostgroup`.

Группа узлов сети

Объект группы узлов сети имеет следующие свойства.

Property	Type	Description
groupid	ID	ID группы узлов сети. Property behavior: - *read-only* - *required* for update operations
name	string	Имя группы узлов сети. Property behavior: - *required* for create operations
flags	integer	Источник группы узлов сети. Possible values: 0 - a plain host group; 4 - a discovered host group. Property behavior: - *read-only*

Property	Type	Description
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных групп узлов сети с уже существующими. Автоматически генерируется, если не указан.

hostgroup.create

Описание

`object hostgroup.create(object/array hostGroups)`

Этот метод позволяет создавать новые группы узлов сети.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Группы узлов сети для создания.

Метод принимает группы узлов сети с стандартными свойствами группы узлов сети.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных групп узлов сети, указанных в свойстве `groupids`. Порядок возвращаемых ID совпадает с порядком переданных групп узлов сети.

Примеры

Создание группы узлов сети

Создайте группу узлов сети с названием ”Linux servers”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

Источник

CHostGroup::create() в *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Описание

`object hostgroup.delete(array hostGroupIds)`

Этот метод позволяет удалять группы узлов сети.

Группа узлов сети не может быть удалена, если:

- она содержит узлы сети, которые принадлежат только этой группе;
- она помечена как внутренняя;
- она используется прототипом узла сети;
- она используется в глобальном скрипте;
- она используется в условии корреляции.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых групп узлов сети.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных групп узлов сети, указанных в свойстве`groupids`.

Примеры

Удаление нескольких групп узлов сети

Удаление двух групп узлов сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

Источник

CHostGroup::delete() в *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Описание

`integer/array hostgroup.get(object parameters)`

Метод позволяет получать группы узлов сети в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
graphids	ID/array	Возвращать только группы узлов сети, которые содержат узлы сети с указанными графиками.
groupids	ID/array	Возвращать только группы узлов сети с указанными идентификаторами групп узлов сети.
hostids	ID/array	Возвращать только группы узлов сети, которые содержат указанные узлы сети.
maintenanceids	ID/array	Возвращать только группы узлов сети, на которые влияют указанные обслуживания.
triggerids	ID/array	Возвращать только группы узлов сети, которые содержат узлы сети с указанными триггерами.
with_graphs	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с графиками. Переопределяет параметр `with_monitored_httptests`.
with_graph_prototypes	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с прототипами графиков. Переопределяет параметр `with_monitored_items`.
with_hosts	boolean	Возвращать только группы узлов сети, которые содержат узлы сети.
with_httptests	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с веб-проверками. Переопределяет параметр `with_monitored_httptests`.
with_items	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с элементами данных. Переопределяет параметры `with_monitored_items` и `with_simple_graph_items`.
with_item_prototypes	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с прототипами элементов данных. Переопределяет параметр `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с прототипами элементов данных, которые включены для создания и имеют числовой тип информации.
with_monitored_httptests	boolean	Возвращать только группы узлов сети, которые содержат контролируемые узлы сети с включенными веб-проверками.
with_monitored_hosts	boolean	Возвращать только группы узлов сети, которые содержат контролируемые узлы сети.
with_monitored_items	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с включенными элементами данных. Переопределяет параметр `with_simple_graph_items`.
with_monitored_triggers	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с включенными триггерами. Все элементы данных, используемые в триггере, также должны быть включены.
with_simple_graph_items	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с числовыми элементами данных.
with_triggers	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с триггерами. Переопределяет параметр `with_monitored_triggers`.
selectDiscoveryRules	query	Возвращать свойство `discoveryRules` с правилами LLD, которые обнаружили группу узлов сети.

Parameter	Type	Description
selectDiscoveryData	query	Возвращать свойство `discoveryData` с объектами обнаружения группы узлов сети. Каждый объект обнаружения группы узлов сети представляет собой прототип группы узлов сети, связанный с обнаруженной группой узлов сети, и имеет следующие свойства: `parent_group_prototypeid` - `(ID)` идентификатор прототипа группы узлов сети, из которого была обнаружена группа узлов сети; `name` - `(string)` имя прототипа группы узлов сети; `ts_delete` - `(timestamp)` время, когда будет удалена группа узлов сети, которая больше не обнаруживается; `status` - `(int)` статус обнаружения группы узлов сети: 0 - *(по умолчанию)* группа узлов сети обнаружена, 1 - группа узлов сети больше не обнаруживается.
selectHostPrototypes	query	Возвращать свойство `hostPrototypes` с прототипами узлов сети, которые обнаружили эту группу узлов сети.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, которые принадлежат группе узлов сети. Поддерживает `count`.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectHosts` - результаты будут отсортированы по `host`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `groupid`, `name`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	Возвращать свойство `groupDiscoveries` с объектами обнаружения группы узлов сети. Каждый объект обнаружения группы узлов сети представляет собой прототип группы узлов сети, связанный с обнаруженной группой узлов сети. Этот запрос **устарел**, вместо него используйте `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов
- количество полученных объектов, если был использован параметр `countOutput`

Примеры

Получение данных по имени

Получить все данные о двух группах узлов сети с именами ”Zabbix servers” и ”Linux servers”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

См. также

- Узел сети

Источник

CHostGroup::get() в *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Описание

`object hostgroup.massadd(object parameters)`

Этот метод позволяет одновременно добавлять несколько связанных объектов во все указанные группы узлов сети.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие IDs групп узлов сети, которые нужно обновить, и объекты, которые нужно добавить во все группы узлов сети.

Метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы узлов сети, которые нужно обновить. Группы узлов сети должны содержать только свойство `groupid`. Поведение параметра: - *required*
hosts	object/array	Узлы сети, которые нужно добавить во все группы узлов сети. Узлы сети должны содержать только свойство `hostid`.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных групп узлов сети, указанных в свойстве `groupids`.

Примеры

Добавление узлов сети в группы узлов сети

Добавьте два узла сети в группы узлов сети с ID 5 и 6.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

См. также

- узел сети

Источник

CHostGroup::massAdd() в *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Описание

`object hostgroup.massremove(object parameters)`

Этот метод позволяет удалить связанные объекты из нескольких групп узлов сети.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы групп узлов сети, которые нужно обновить, и объекты, которые следует удалить.

Parameter	Type	Description
groupids	ID/array	Идентификаторы групп узлов сети, которые нужно обновить. Parameter behavior: - *required*
hostids	ID/array	Идентификаторы узлов сети, которые нужно удалить из всех групп узлов сети.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных групп узлов сети, указанных в свойстве `groupids`.

Примеры

Удаление узлов сети из групп узлов сети

Удалите два узла сети из указанных групп узлов сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Источник

CHostGroup::massRemove() в *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Описание

`object hostgroup.propagate(object parameters)`

Этот метод позволяет применять разрешения и фильтры тегов ко всем подгруппам группы узлов сети.

> **Note:**
> Этот метод доступен только для типов пользователей *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

|Parameter|Type|Description|

|--|--|------|

|groups|object/array|Группы узлов сети для распространения.

Группы узлов сети должны иметь заданное свойство `groupid`.

Поведение параметра:
- *required*|

|permissions|boolean|Установите значение "true", чтобы распространить разрешения.

Поведение параметра:
- *required* если `tag_filters` не задан|

|tag_filters|boolean|Установите значение "true", чтобы распространить фильтры тегов.

Поведение параметра:
- *required* если `permissions` не задан|

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы распространенных групп узлов сети в свойстве `groupids`.

Примеры

Распространение прав доступа к группе узлов сети и фильтров тегов на ее подгруппы.

Распространить права доступа к группе узлов сети и фильтры тегов на ее подгруппы.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
```

```
    "id": 1
}
```

См. также

- hostgroup.update
- hostgroup.massadd
- Узел сети

Источник

CHostGroup::propagate() в

ui/include/classes/api/services/CHostGroup.php.

hostgroup.update

Описание

`object hostgroup.update(object/array hostGroups)`

Этот метод позволяет обновлять существующие группы узлов сети.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства группы узлов сети, которые необходимо обновить.

Свойство `groupid` должно быть определено для каждой группы узлов сети, все остальные свойства необязательны. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных групп узлов сети, указанных в свойстве `groupids`.

Примеры

Переименование группы узлов сети

Переименуйте группу узлов сети в ”Linux hosts”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Переименование нескольких групп узлов сети

Переименуйте две группы узлов сети, чтобы включить информацию о регионе в их имена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Источник

CHostGroup::update() в *ui/include/classes/api/services/CHostGroup.php*.

Группа шаблонов

Этот класс предназначен для работы с группами шаблонов.

Ссылки на объекты:

- Группа шаблонов

Доступные методы:

- templategroup.create - создать новые группы шаблонов
- templategroup.delete - удалить группы шаблонов
- templategroup.get - получить группы шаблонов
- templategroup.massadd - добавить связанные объекты в группы шаблонов
- templategroup.massremove - удалить связанные объекты из групп шаблонов
- templategroup.propagate - распространить права доступа на подгруппы групп шаблонов
- templategroup.update - обновить группы шаблонов

Объект группы шаблонов

Следующие объекты напрямую связаны с API `templategroup`.

Группа шаблонов

Объект группы шаблонов имеет следующие свойства.

Property	Type	Description
groupid	ID	ID группы шаблонов. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя группы шаблонов. Поведение свойства: - *обязательно* для операций создания
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных групп шаблонов с уже существующими. Генерируется автоматически, если не указан.

templategroup.create

Описание

`object templategroup.create(object/array templateGroups)`

Этот метод позволяет создавать новые группы шаблонов.

> **Note:**
> Этот метод доступен только пользователю типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Группы шаблонов для создания. Метод принимает группы шаблонов с стандартными свойствами группы шаблонов.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных групп шаблонов в свойстве `groupids`. Порядок возвращаемых идентификаторов соответствует порядку переданных групп шаблонов.

Примеры

Создание группы шаблонов

Создайте группу шаблонов с именем "Templates/Databases".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

Источник

CTemplateGroup::create() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Описание

`object templategroup.delete(array templateGroupIds)`

Этот метод позволяет удалять группы шаблонов.

Группу шаблонов нельзя удалить, если она содержит шаблоны, которые принадлежат только этой группе.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы групп шаблонов, которые необходимо удалить.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленного шаблона. группы под свойством `groupids`.

Примеры

Удаление нескольких групп шаблонов

Удаление двух групп шаблонов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Источник

CTemplateGroup::delete() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Описание

`integer/array templategroup.get(object parameters)`

Метод позволяет получать группы шаблонов в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
graphids	ID/array	Возвращать только группы шаблонов, которые содержат шаблоны с указанными графиками.
groupids	ID/array	Возвращать только группы шаблонов с указанными идентификаторами групп шаблонов.
templateids	ID/array	Возвращать только группы шаблонов, которые содержат указанные шаблоны.
triggerids	ID/array	Возвращать только группы шаблонов, которые содержат шаблоны с указанными триггерами.
with_graphs	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с графиками. Переопределяет параметры `with_simple_graph_items`.
with_graph_prototypes	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с прототипами графиков.
with_httptests	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с веб-проверками.
with_items	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с элементами данных. Переопределяет параметр `with_simple_graph_items`.
with_item_prototypes	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с прототипами элементов данных. Переопределяет параметр `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с прототипами элементов данных, включенными для создания и имеющими числовой тип информации.
with_simple_graph_items	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с числовыми элементами данных.
with_templates	boolean	Возвращать только группы шаблонов, которые содержат шаблоны.
with_triggers	boolean	Возвращать только группы шаблонов, которые содержат шаблоны с триггерами.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, которые принадлежат группе шаблонов. Поддерживает `count`.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectTemplates` - результаты будут отсортированы по `template`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `groupid`, `name`.
countOutput	boolean	Эти параметры описаны в комментарии к справочнику.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parameter	Type	Description
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение данных по имени

Получить все данные о двух группах шаблонов с именами "Templates/Databases" и "Templates/Modules".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

См. также

- Шаблон

Исходный код

CTemplateGroup::get() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Описание

`object templategroup.massadd(object parameters)`

Этот метод позволяет одновременно добавлять несколько связанных объектов во все указанные группы шаблонов.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, содержащие идентификаторы групп шаблонов для обновления и объекты, которые нужно добавить во все группы шаблонов.

Метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы шаблонов, которые нужно обновить. Группы шаблонов должны содержать только свойство `groupid`. Поведение параметра: - *required*
templates	object/array	Шаблоны, которые нужно добавить во все группы шаблонов. Шаблоны должны содержать только свойство `templateid`. Поведение параметра: - *required*

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных групп шаблонов в свойстве `groupids`.

Примеры

Добавление шаблонов в группы шаблонов

Добавьте два шаблона в группы шаблонов с ID 12 и 13.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

См. также

- Шаблон

Источник

CTemplateGroup::massAdd() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Описание

`object templategroup.massremove(object parameters)`

Этот метод позволяет удалять связанные объекты из нескольких групп шаблонов.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object)` Параметры, содержащие ID групп шаблонов, которые нужно обновить, и объекты, которые следует удалить.

Parameter	Type	Description
groupids	ID/array	ID групп шаблонов, которые нужно обновить. Поведение параметра: - *required*
templateids	ID/array	ID шаблонов, которые нужно удалить из всех групп шаблонов. Поведение параметра: - *required*

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных групп шаблонов в свойстве `groupids`.

Примеры

Удаление шаблонов из групп шаблонов

Удалите два шаблона из указанных групп шаблонов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Источник

CTemplateGroup::massRemove() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Описание

`object templategroup.propagate(object parameters)`

Этот метод позволяет применять разрешения ко всем подгруппам групп шаблонов.

Note:
Этот метод доступен только для типов пользователей *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

|Parameter|Type|Description|

|--|--|------|

|groups|object/array|Группы шаблонов для распространения.

Группы шаблонов должны содержать только свойство `groupid`.

Поведение параметра:
- *required*|

|permissions|boolean|Установите `true`, если необходимо распространить права доступа.

Поведение параметра:
- *required*|

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы распространенных групп шаблонов в свойстве `groupids`.

Примеры

Распространение прав доступа группы шаблонов на ее подгруппы.

Распространите права доступа группы шаблонов на ее подгруппы.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

См. также

- templategroup.update
- templategroup.massadd
- Шаблон

Исходный код

CTemplateGroup::propagate() в

ui/include/classes/api/services/CTemplateGroup.php.

templategroup.update

Описание

`object templategroup.update(object/array templateGroups)`

Этот метод позволяет обновлять существующие группы шаблонов.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства группы шаблонов, которые необходимо обновить.

Свойство `groupid` должно быть определено для каждой группы шаблонов, все остальные свойства необязательны. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных групп шаблонов в свойстве `groupids`.

Примеры

Переименование группы шаблонов

Переименовать группу шаблонов в ”Templates/Databases”

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Исходный код

CTemplateGroup::update() в *ui/include/classes/api/services/CTemplateGroup.php*.

Журнал APM

Этот класс предназначен для работы с журналами, собранными для мониторинга производительности приложений (APM).

Ссылки на объекты:

- Журнал APM

Доступные методы:

- apmlog.get - получение записей журнала

Объект журнала APM

Следующие объекты напрямую связаны с API `apmlog`.

Note:
Каждое свойство соответствует столбцу с таким же именем в таблице `otel_logs` источника данных телеметрии (например, свойство `severity_text` соответствует столбцу `SeverityText`). Список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии.

Журнал APM

Объект журнала APM представляет запись журнала (текстовую запись события, например неудачного платежа или тайм-аута базы данных). Запись журнала содержит информацию о событии, произошедшем в определённый момент времени, включая его важность, сообщение и атрибуты. Если запись журнала была создана во время операции, являющейся частью трассировки, она также содержит идентификаторы этой трассировки и спана.

Объект журнала APM имеет следующие свойства:

Свойство	Тип	Описание
timestamp	integer	Время в формате Unix (в наносекундах), когда произошло событие. Поведение свойства: - *только для чтения*
traceid	string	Идентификатор трассировки, во время которой была создана запись журнала, если он имеется. Поведение свойства: - *только для чтения*
spanid	string	Идентификатор спана, во время которого была создана запись журнала, если он имеется. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
trace_flags	integer	Флаги трассировки контекста трассировки записи журнала в виде битовой маски. Если установлен младший значащий бит (значение 1), трассировка подвергается выборке. Поведение свойства: - *только для чтения*
severity_text	string	Уровень важности записи журнала, указанный приложением (например, `INFO`, `WARN` или `ERROR`). Поведение свойства: - *только для чтения*
severity_number	integer	Уровень важности записи журнала в виде числа. Возможные значения: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL. Поведение свойства: - *только для чтения*
service_name	string	Имя сервиса, создавшего запись журнала (например, `zabbix-frontend`), полученное из атрибута ресурса OpenTelemetry `service.name`. Поведение свойства: - *только для чтения*
body	string	Сообщение записи журнала. Поведение свойства: - *только для чтения*
resource_schema_url	string	URL схемы OpenTelemetry атрибутов ресурса, если он имеется. Поведение свойства: - *только для чтения*
resource_attributes	object	Пары ключ-значение атрибутов ресурса записи журнала (например, `host.name: eabf8c2a5ddc`). Поведение свойства: - *только для чтения*
scope_schema_url	string	URL схемы OpenTelemetry области инструментирования и её атрибутов, если он имеется. Поведение свойства: - *только для чтения*
scope_name	string	Имя области инструментирования, создавшей запись журнала. Поведение свойства: - *только для чтения*
scope_version	string	Версия области инструментирования, если она имеется. Поведение свойства: - *только для чтения*
scope_attributes	object	Атрибуты области инструментирования в формате ключ-значение. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
log_attributes	object	Пары ключ-значение атрибутов журнала записи журнала (например, `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Поведение свойства: - *только для чтения*
event_name	string	Имя типа события, представленного записью журнала, если оно имеется. Поведение свойства: - *только для чтения*

apmlog.get

Описание

`integer/array apmlog.get(object parameters)`

Метод позволяет получать записи журнала в соответствии с указанными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
time_from	timestamp	Возвращать только записи журнала событий, произошедших в указанное время или позднее. Поведение параметра: - *обязательный*
time_till	timestamp	Возвращать только записи журнала событий, произошедших до указанного времени. Поведение параметра: - *обязательный*
traceids	string/array	Возвращать только записи журнала, созданные во время указанных трассировок.
spanids	string/array	Возвращать только записи журнала, созданные во время указанных интервалов.
with_flags_on	integer	Возвращать только записи журнала, у которых установлены все указанные флаги трассировки. Чтобы указать несколько флагов, используйте сумму их значений.
with_flags_off	integer	Возвращать только записи журнала, у которых не установлен ни один из указанных флагов трассировки. Чтобы указать несколько флагов, используйте сумму их значений.

Параметр	Тип	Описание
resource_attributes	object/array	Возвращать только записи журнала с указанными атрибутами ресурса. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все записи журнала. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
resource_attributes_evaltype	integer	Метод оценки условий в `resource_attributes`. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.
scope_attributes	object/array	Возвращать только записи журнала с указанными атрибутами области действия. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все записи журнала. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
scope_attributes_evaltype	integer	Метод оценки условий в `scope_attributes`. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.
log_attributes	object/array	Возвращать только записи журнала с указанными атрибутами журнала. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все записи журнала. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
log_attributes_evaltype	integer	Метод оценки условий в `log_attributes`. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.

Параметр	Тип	Описание
filter	object	Возвращать только результаты, которые точно соответствуют указанному фильтру. Принимает объект, в котором ключами являются имена свойств, а значениями - отдельные значения или массивы значений для сопоставления. Поддерживаемые свойства: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, в котором ключами являются имена свойств, а значениями - строки для поиска. Если дополнительные параметры не указаны, выполняется поиск `LIKE "%…%"`. Поддерживаемые свойства: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Количество записей, которые необходимо пропустить перед возвратом результата.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение журналов ошибок трассировки

Получите записи журнала с уровнем серьезности `ERROR` (номера уровней серьезности 17-20), связанные с трассировкой "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
```

```
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

Поиск сообщений в журналах

Получение времени, службы и сообщения 10 последних записей журнала, содержащих в сообщении слово ”timeout”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```


См. также

- Трассировка APM
- Сегмент APM
- Метрика APM

Источник

CApmLog::get() в файле *ui/include/classes/api/services/CApmLog.php*.

Журнал аудита

Этот класс предназначен для работы с журналом аудита.

Ссылки на объекты:

- Журнал аудита

Доступные методы:

- auditlog.get - получить записи журнала аудита

Объект журнала аудита

Следующие объекты напрямую связаны с API `auditlog`.

Журнал аудита

Объект журнала аудита содержит информацию о действиях пользователей. Он имеет следующие свойства.

Property	Type	Description
auditid	ID	ID записи журнала аудита. Сгенерирован с использованием алгоритма CUID.
userid	ID	ID пользователя-автора записи журнала аудита.
username	string	Имя пользователя-автора записи журнала аудита.
clock	timestamp	Метка времени создания записи журнала аудита.
ip	string	IP-адрес пользователя-автора записи журнала аудита.

Property	Type	Description
action	integer	Действие записи журнала аудита. Возможные значения: 0 - Добавление; 1 - Обновление; 2 - Удаление; 4 - Выход; 7 - Выполнение; 8 - Вход; 9 - Неудачный вход; 10 - Очистка истории; 11 - Обновление конфигурации; 12 - Отправка.

Property	Type	Description
resourcetype	integer	Тип ресурса записи журнала аудита. Возможные значения: 0 - Пользователь; 3 - Тип медиа; 4 - Узел сети; 5 - Действие; 6 - График; 11 - Группа пользователей; 13 - Триггер; 14 - Группа узлов сети; 15 - Элемент данных; 16 - Изображение; 17 - Карта значений; 18 - Сервис; 19 - Карта; 22 - Веб-сценарий; 23 - Правило обнаружения; 25 - Скрипт; 26 - Прокси; 27 - Обслуживание; 28 - Регулярное выражение; 29 - Макрос; 30 - Шаблон; 31 - Прототип триггера; 32 - Сопоставление значков; 33 - Панель мониторинга; 34 - Корреляция событий; 35 - Прототип графика; 36 - Прототип элемента данных; 37 - Прототип узла сети; 38 - Автоматическая регистрация; 39 - Модуль; 40 - Настройки; 41 - Обслуживание; 42 - Аутентификация; 43 - Панель мониторинга шаблона; 44 - Роль пользователя; 45 - Токен API; 46 - Запланированный отчет; 47 - Узел высокой доступности; 48 - SLA; 49 - Каталог пользователей; 50 - Группа шаблонов; 51 - Коннектор; 52 - Правило LLD; 53 - История; 54 - Многофакторная аутентификация; 55 - Группа прокси; 56 - Прототип правила LLD.
resourceid	ID	Идентификатор ресурса записи журнала аудита.
resource_cuid	ID	Уникальный идентификатор ресурса записи журнала аудита, сгенерированный с использованием алгоритма CUID.
resourcename	string	Читаемое имя ресурса записи журнала аудита.
recordsetid	ID	ID набора записей записи журнала аудита. Записи журнала аудита, созданные в ходе одной и той же операции, будут иметь одинаковый ID набора записей. Сгенерирован с использованием алгоритма CUID.

Property	Type	Description
details	text	Подробности записи журнала аудита. Подробности хранятся в виде объекта JSON, где каждое имя свойства является путем к свойству или вложенному объекту, в котором произошло изменение, а каждое значение содержит данные (в формате массива) об изменении в этом свойстве или вложенном объекте. Возможные форматы значений: ["add"] - Вложенный объект был добавлен; ["add", "<value>"] - Свойство добавленного объекта равно <value>; ["update"] - Вложенный объект был обновлен; ["update", "<new value>", "<old value>"] - Свойство обновленного объекта было изменено с <old value> на <new value>; ["delete"] - Вложенный объект был удален.

auditlog.get

Описание

`integer/array auditlog.get(object parameters)`

Метод позволяет получать записи журнала аудита в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
auditids	ID/array	Возвращать только журнал аудита с указанными ID.
userids	ID/array	Возвращать только журнал аудита, созданный указанными пользователями.
time_from	timestamp	Возвращать только записи журнала аудита, созданные после указанного времени или в указанное время.
time_till	timestamp	Возвращать только записи журнала аудита, созданные до указанного времени или в указанное время.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `auditid`, `userid`, `clock`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;

- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение журнала аудита

Получить два последних записи журнала аудита.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resource_cuid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resource_cuid": "0",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

Смотрите также

- ·Audit log object

Источник

CAuditLog::get() в *ui/include/classes/api/services/CAuditLog.php*.

Задача

Этот класс предназначен для работы с задачами (например, для проверки элементов данных или правил низкоуровневого обнаружения без перезагрузки конфигурации).

Ссылки на объекты:

- Task
  - объект запроса 'Execute now'
  - объект запроса 'Refresh proxy configuration'
  - объект запроса 'Diagnostic information'
    * объект запроса статистики
    * объект результата статистики

Доступные методы:

- task.create - создать новые задачи
- task.get - получить задачи

Объект task

Следующие объекты напрямую связаны с API `task`.

Task

Объект task имеет следующие свойства:

Property	Type	Description
taskid	ID	ID задачи. Поведение свойства: - *только для чтения*
type	integer	Тип задачи. Возможные значения: 1 - Диагностическая информация; 2 - Обновление конфигурации прокси; 6 - Выполнить сейчас. Пользователи типов *Admin* и *User* могут создавать только задачи типа "Выполнить сейчас". Обратите внимание, что `task.get` всегда возвращает "7" (сводка выполнения задачи). Поведение свойства: - *обязательно*
status	integer	Статус задачи. Возможные значения: 1 - новая задача; 2 - задача в процессе выполнения; 3 - задача завершена; 4 - срок действия задачи истек. Поведение свойства: - *только для чтения*
clock	timestamp	Время создания задачи. Поведение свойства: - *только для чтения*

Property	Type	Description
ttl	integer	Время в секундах, по истечении которого задача становится просроченной. Поведение свойства: - *только для чтения*
proxyid	ID	ID прокси, по которому собирается статистика диагностической информации. Поведение свойства: - *поддерживается*, если `type` установлено в ”Диагностическая информация” или ”Обновление конфигурации прокси”
request	object	Объект запроса задачи в соответствии с типом задачи: Объект задачи ”Выполнить сейчас” подробно описан ниже; Объект задачи ”Обновление конфигурации прокси” подробно описан ниже; Объект задачи ”Диагностическая информация” подробно описан ниже. Поведение свойства: - *обязательно*
result	object	Объект результата задачи диагностической информации. Может содержать NULL, если результат еще не готов. Объект результата подробно описан ниже. Поведение свойства: - *только для чтения*

Объект запроса ’Execute now’

Объект запроса задачи ’Execute now’ имеет следующие свойства.

Property	Type	Description
itemid	ID	ID элемента данных и правил низкоуровневого обнаружения. *Admin* и *User* пользователи могут выполнять ’Execute now’ для элементов данных на узлах сети, для которых у них есть разрешение *read-write*, либо разрешение *read* и для их роли включено действие `invoke_execute_now`. То же самое относится к пользователям типа *Admin* для правил низкоуровневого обнаружения (LLD).

Объект запроса ’Refresh proxy configuration’

Объект запроса задачи ’Refresh proxy configuration’ имеет следующие свойства.

Property	Type	Description
proxyids	array	Идентификаторы прокси.

Объект запроса 'Diagnostic information'

Объект запроса задачи диагностической информации имеет следующие свойства. Объект запроса статистики для всех типов свойств подробно описан ниже.

Property	Type	Description
historycache	object	Запрос статистики кэша истории. Доступно на сервере и прокси.
valuecache	object	Запрос статистики кэша элементов данных. Доступно на сервере.
preprocessing	object	Запрос статистики менеджера предварительной обработки. Доступно на сервере и прокси.
alerting	object	Запрос статистики менеджера оповещений. Доступно на сервере.
lld	object	Запрос статистики менеджера LLD. Доступно на сервере.

Объект запроса статистики

Объект запроса статистики используется для определения того, какой тип информации следует собирать о внутренних процессах сервера/прокси. Он имеет следующие свойства.

Property	Type	Description
stats	query	Свойства объекта статистики, которые нужно вернуть. Список доступных полей для каждого типа статистики диагностической информации подробно описан ниже. По умолчанию: `extend` вернет все доступные поля статистики.
top	object	Объект для сортировки и ограничения возвращаемых значений статистики. Список доступных полей для каждого типа статистики диагностической информации подробно описан ниже. Пример: { "source.alerts": 10 }

Список полей статистики, доступных для каждого типа запроса диагностической информации

Для каждого свойства запроса диагностической информации можно запросить следующие поля статистики.

Diagnostic type	Available fields	Description
historycache	items	Количество кэшированных элементов данных.
	values	Количество кэшированных значений.
	memory	Статистика общей памяти (свободное пространство, количество используемых блоков, количество свободных блоков, максимальный размер свободного блока).
	memory.data	Статистика общей памяти кэша исторических данных.
	memory.index	Статистика общей памяти кэша индекса истории.
valuecache	items	Количество кэшированных элементов данных.
	values	Количество кэшированных значений.
	memory	Статистика общей памяти (свободное пространство, количество используемых блоков, количество свободных блоков, максимальный размер свободного блока).
	mode	Режим кэша значений.
preprocessing	values	Количество значений в очереди.
	preproc.values	Количество значений в очереди с шагами предварительной обработки.
alerting	alerts	Количество оповещений в очереди.
lld	rules	Количество правил в очереди.
	values	Количество значений в очереди.

Список полей сортировки, доступных для каждого типа запроса диагностической информации

Следующие статистические поля можно использовать для сортировки и ограничения запрашиваемой информации.

Diagnostic type	Available fields	Type
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Объект результата статистики

Объект результата статистики извлекается из поля `result` объекта задачи.

Property	Type	Description
status	integer	Статус результата задачи. Возможные значения: -1 - при выполнении задачи произошла ошибка; 0 - результат задачи создан. Поведение свойства: - *только для чтения*
data	string/object	Результаты в соответствии с объектом запроса статистики для конкретной задачи диагностической информации. Содержит строку сообщения об ошибке, если при выполнении задачи произошла ошибка.

task.create

Описание

`object task.create(object/array tasks)`

Этот метод позволяет создавать задачи.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Задача для создания.

Метод принимает задачи со стандартными свойствами задачи.

Обратите внимание, что задачи типа "Выполнить сейчас" можно создавать только для следующих типов элементов данных/правил обнаружения:

- Zabbix agent (passive)
- Simple check
- SNMP agent (v1/v2/v3)
- Zabbix internal
- External check
- Database monitor
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Calculated
- Dependent item
- Script
- Browser

Если элемент данных/правило обнаружения имеет тип ”Dependent item”, то его мастер-элемент данных также должен относиться к одному из указанных выше типов.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных задач, указанных в свойстве `taskids`. Одна задача создается по каждому элементу данных и правилу низкоуровневого обнаружения. Порядок возвращаемых ID совпадает с порядком переданных `itemids`.

Примеры

Создание задачи

Создайте задачу «Выполнить сейчас» для элемента данных и правила низкоуровневого обнаружения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Создайте задачу «Обновить конфигурацию прокси» для двух прокси.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Создайте задачу «Диагностическая информация».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 1,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "proxyid": 0
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

См. также

- Задача
- Объект запроса 'Execute now'
- Объект запроса 'Refresh proxy configuration'
- Объект запроса 'Diagnostic information'

Источник

CTask::create() в *ui/include/classes/api/services/CTask.php*.

task.get

Описание

`integer/array task.get(object parameters)`

Метод позволяет получать задачи в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
taskids	ID/array	Возвращать только задачи с указанными ID.
output	query	Эти параметры описаны в справочном комментарии.
preservekeys	boolean	

Возвращаемые значения

`(` `/` `)` Возвращает массив объектов.

Примеры

Получение задачи по ID

Получите все данные о задаче с ID "1".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "taskid": "1",
            "type": "7",
            "status": "3",
            "clock": "1601039076",
            "ttl": "3600",
            "proxyid": null,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
```

```
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "result": {
                "data": {
                    "alerting": {
                        "alerts": 0,
                        "top": {
                            "media.alerts": []
                        },
                        "time": 0.000663
                    },
                    "lld": {
                        "rules": 0,
                        "values": 0,
                        "top": {
                            "values": []
                        },
                        "time": 0.000442
                    }
                },
                "status": "0"
            }
        }
    ],
    "id": 1
}
```


См. также

- Задача
- Объект запроса 'Execute now'
- Объект запроса 'Refresh proxy configuration'
- Объект запроса 'Diagnostic information'

Источник

CTask::get() в *ui/include/classes/api/services/CTask.php*.

Изображение

Этот класс предназначен для работы с изображениями.

Ссылки на объекты:

- Image

Доступные методы:

- image.create - создать новые изображения
- image.delete - удалить изображения
- image.get - получить изображения
- image.update - обновить изображения

Объект image

Следующие объекты напрямую связаны с API `image`.

Изображение

Объект image имеет следующие свойства.

Property	Type	Description
imageid	ID	ID изображения. Property behavior: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя изображения. Property behavior: - *обязательно* для операций создания
imagetype	integer	Тип изображения. Возможные значения: 1 - *(по умолчанию)* значок; 2 - фоновое изображение. Property behavior: - *константа* - *обязательно* для операций создания
image	string	Изображение, закодированное в Base64. Максимальный размер закодированного изображения — 1 MB. Максимальный размер можно изменить, изменив значение константы ZBX_MAX_IMAGE_SIZE. Поддерживаемые форматы изображений: PNG, JPEG, GIF и WebP. Property behavior: - *обязательно* для операций создания

image.create

Описание

`object image.create(object/array images)`

Этот метод позволяет создавать новые изображения.

Note:
Этот метод доступен только пользователю типа *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Изображения для создания.

Метод принимает изображения со стандартными свойствами изображения.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных изображений, указанных в свойстве `imageids`. Порядок возвращаемых ID совпадает с порядком переданных изображений.

Примеры

Создание изображения значка

Создайте значок облака.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

Создание фонового изображения

Создайте небольшое фоновое изображение, которое будет использоваться в качестве шаблона.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Источник

CImage::create() в *ui/include/classes/api/services/CImage.php*.

image.delete

Описание

`object image.delete(array imageIds)`

Этот метод позволяет удалять изображения.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых изображений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных изображений, указанных в свойстве `imageids`.

Примеры

Удаление нескольких изображений

Удаление двух изображений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Источник

CImage::delete() в *ui/include/classes/api/services/CImage.php*.

image.get

Описание

`integer/array image.get(object parameters)`

Метод позволяет получать изображения в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
imageids	ID/array	Возвращать только изображения с указанными ID.
sysmapids	ID/array	Возвращать изображения, которые используются на указанных картах.
select_image	flag	Возвращать свойство `image` с изображением, закодированным в Base64.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `imageid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение изображения

Получить все данные для изображения с ID "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Источник

CImage::get() в *ui/include/classes/api/services/CImage.php*.

image.update

Описание

`object image.update(object/array images)`

Этот метод позволяет обновлять существующие изображения.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object/array) Свойства изображения, которые необходимо обновить.

Свойство imageid должно быть определено для каждого изображения, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Метод принимает изображения со стандартными свойствами изображения.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных изображений, указанных в свойстве imageids.

Примеры

Переименовать изображение

Переименуйте изображение в "Cloud icon".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Источник

CImage::update() в *ui/include/classes/api/services/CImage.php*.

Интерфейс узла сети

Этот класс предоставляет доступ к интерфейсам узла сети.

Ссылки на объекты:

- Интерфейс узла сети
  - Подробности

Доступный метод:

- hostinterface.get - получение интерфейсов узла сети

Интерфейсы узла сети создаются, обновляются и удаляются с помощью методов узла сети.

Объект интерфейса узла сети

Следующие объекты напрямую связаны с API hostinterface.

Интерфейс узла сети

Объект интерфейса узла сети имеет следующие свойства.

Свойство	Тип	Описание
interfaceid	ID	Идентификатор интерфейса. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
available	integer	Доступность интерфейса узла сети. Возможные значения: 0 - *(по умолчанию)* неизвестно; 1 - доступен; 2 - недоступен. Поведение свойства: - *только для чтения*
hostid	ID	Идентификатор узла сети, которому принадлежит интерфейс. Поведение свойства: - *константа* - *обязательно* для операций создания
type	integer	Тип интерфейса. Возможные значения: 1 - агент; 2 - SNMP; 3 - IPMI; 4 - JMX. Поведение свойства: - *обязательно* для операций создания
ip	string	IP-адрес, используемый интерфейсом. Может быть пустым, если подключение выполняется через DNS. Поведение свойства: - *обязательно* для операций создания
dns	string	Имя DNS, используемое интерфейсом. Может быть пустым, если подключение выполняется через IP. Поведение свойства: - *обязательно* для операций создания
port	string	Номер порта, используемый интерфейсом. Может содержать пользовательские макросы. Поведение свойства: - *обязательно* для операций создания
useip	integer	Определяет, должно ли подключение выполняться через IP. Возможные значения: 0 - подключаться, используя имя узла сети в DNS; 1 - подключаться, используя IP-адрес узла сети. Поведение свойства: - *обязательно* для операций создания

Свойство	Тип	Описание
main	integer	Определяет, используется ли интерфейс по умолчанию на узле сети. На узле сети только один интерфейс определенного типа может быть установлен как используемый по умолчанию. Возможные значения: 0 - не используется по умолчанию; 1 - используется по умолчанию. Поведение свойства: - *обязательно* для операций создания
details	object	Дополнительный объект details для интерфейса. Поведение свойства: - *обязательно*, если для `type` установлено значение "SNMP"
disable_until	timestamp	Время следующего опроса недоступного интерфейса узла сети. Поведение свойства: - *только для чтения*
error	string	Текст ошибки, если интерфейс узла сети недоступен. Поведение свойства: - *только для чтения*
errors_from	timestamp	Время, когда интерфейс узла сети стал недоступен. Поведение свойства: - *только для чтения*

Подробности

Объект details имеет следующие свойства.

Property	Type	Description
version	integer	Версия SNMP-интерфейса. Возможные значения: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Поведение свойства: - *required*
bulk	integer	Использовать ли массовые SNMP-запросы. Возможные значения: 0 - не использовать массовые запросы; 1 - (по умолчанию) - использовать массовые запросы.
community	string	SNMP community. Используется только интерфейсами SNMPv1 и SNMPv2. Поведение свойства: - *required* если `version` установлено в "SNMPv1" или "SNMPv2c"
max_repetitions	integer	Максимальное значение повторений для нативных массовых SNMP-запросов (PDU GetBulkRequest). Используется только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. По умолчанию: 10.
securityname	string	Имя безопасности SNMPv3. Используется только интерфейсами SNMPv3.

Property	Type	Description
securitylevel	integer	Уровень безопасности SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - (по умолчанию) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	Фраза-пароль аутентификации SNMPv3. Используется только интерфейсами SNMPv3.
privpassphrase	string	Фраза-пароль конфиденциальности SNMPv3. Используется только интерфейсами SNMPv3.
authprotocol	integer	Протокол аутентификации SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - (по умолчанию) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	Протокол конфиденциальности SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - (по умолчанию) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	Имя контекста SNMPv3. Используется только интерфейсами SNMPv3.

hostinterface.get

Описание

`integer/array hostinterface.get(object parameters)`

Метод позволяет получить интерфейсы узлов сети в соответствии с указанными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
hostids	ID/array	Возвращать только интерфейсы узлов сети, используемые указанными узлами сети.
interfaceids	ID/array	Возвращать только интерфейсы узлов сети с указанными ID.
itemids	ID/array	Возвращать только интерфейсы узлов сети, используемые указанными элементами данных.
triggerids	ID/array	Возвращать только интерфейсы узлов сети, используемые элементами данных в указанных триггерах.

Parameter	Type	Description
selectItems	query	Возвращать свойство `items` с элементами данных, которые используют интерфейс. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, которые используют интерфейс.
limitSelects	integer	Ограничивает количество записей, возвращаемых подвыборками. Применяется к следующим подвыборкам: `selectItems`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение интерфейсов узла сети

Получить все данные об интерфейсах, используемых узлом сети ”30057.”

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "interfaceid": "50039",
            "hostid": "30057",
            "main": "1",
            "type": "1",
            "useip": "0",
            "ip": "",
            "dns": "localhost",
            "port": "10050",
```

```
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": []
        },
        {
            "interfaceid": "55082",
            "hostid": "30057",
            "main": "1",
            "type": "2",
            "useip": "1",
            "ip": "127.0.0.1",
            "dns": "",
            "port": "161",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": {
                "version": "2",
                "bulk": "0",
                "community": "{$SNMP_COMMUNITY}",
                "max_repetitions": "10"
            }
        }
    ],
    "id": 1
}
```


Смотрите также

- Узел сети
- Элемент данных

Источник

CHostInterface::get() в *ui/include/classes/api/services/CHostInterface.php*.

Информация о сервере

Этот класс предназначен для получения информации о сервере Zabbix.

Доступные методы:

- serverinfo.get - получение времени последнего обновления с сервера Zabbix

serverinfo.get

Этот метод позволяет получить время последнего обновления указанного сервера Zabbix.

Метод доступен пользователям с ролями *User*, *Admin*, *Super admin*, а также неаутентифицированным пользователям.

Сигнатура метода: `object serverinfo.get(object parameters)`

Исходный код метода: `CServerInfo::get()` в `ui/include/classes/api/services/CServerInfo.php`

Параметры

Parameter	Type	Description
serverid	ID	Идентификатор сервера Zabbix. Поведение параметра: - Обязательный
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

Свойство	Тип	Описание
serverid	ID	Идентификатор сервера Zabbix.
lastaccess	timestamp	Время последнего обновления от сервера Zabbix. Для кластера высокой доступности (HA) отображается значение `lastaccess` активного узла.

Примеры

Получение информации о сервере

Получение информации об указанном сервере Zabbix.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

История

Этот класс предназначен для работы с данными истории.

Ссылки на объекты:

- История float
- История integer
- История string
- История text
- История log

Доступные методы:

- history.clear - очистить данные истории
- history.get - получить данные истории
- history.push - отправить данные истории на сервер Zabbix

Объект history

Следующие объекты напрямую связаны с API `history`.

> **Note:**
> Объекты history различаются в зависимости от типа информации элемента данных. Они создаются сервером Zabbix и не могут быть изменены через API.

История float

Объект истории float имеет следующие свойства.

Property	Type	Description
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда было получено значение.
value	float	Полученное значение.

История целых чисел

Объект истории целых чисел имеет следующие свойства.

Property	Type	Description
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда было получено значение.
value	integer	Полученное значение.

История строк

Объект истории строк имеет следующие свойства.

Property	Type	Description
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда было получено значение.
value	string	Полученное значение.

История текста

Объект истории текста имеет следующие свойства.

Property	Type	Description
id	ID	ID записи истории.
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда было получено значение.
value	text	Полученное значение.

История журнала

Объект истории журнала имеет следующие свойства.

Property	Type	Description
id	ID	ID записи истории.
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
logeventid	integer	ID записи журнала событий Windows.
ns	integer	Наносекунды, когда было получено значение.
severity	integer	Уровень записи журнала событий Windows.
source	string	Источник записи журнала событий Windows.
timestamp	timestamp	Время записи журнала событий Windows.
value	text	Полученное значение.

Двоичная история

Объект двоичной истории имеет следующие свойства.

Property	Type	Description
clock	timestamp	Время, когда было получено это значение.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда было получено значение.
value	text	Полученное значение.

JSON history

Объект истории JSON имеет следующие свойства.

Свойство	Тип	Описание
clock	timestamp	Время, когда это значение было получено.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда значение было получено.
value	text	Полученное значение.

history.clear

Описание

`object history.clear(array itemids)`

Этот метод позволяет очистить историю элемента данных.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` ID элементов для очистки.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы очищенных элементов в свойстве `itemids`.

Примеры

Очистка истории

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

Источник

CHistory::clear() в *ui/include/classes/api/services/CHistory.php*.

history.get

Описание

`integer/array history.get(object parameters)`

Метод позволяет получать данные истории в соответствии с заданными параметрами.

Attention:
Этот метод может вернуть исторические данные удаленной сущности, если эти данные еще не были удалены housekeeper.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
history	integer	Типы объектов истории, которые нужно вернуть. Возможные значения: 0 - числовое число с плавающей точкой; 1 - символ; 2 - журнал; 3 - *(по умолчанию)* числовое беззнаковое; 4 - текст; 5 - двоичный; 6 - JSON.
hostids	ID/array	Возвращать только историю с указанных узлов сети.
itemids	ID/array	Возвращать только историю с указанных элементов данных.
maxValueSize	integer	Максимальное число байт, которое нужно вернуть в свойстве `value`. Возможный диапазон значений: 1-134217728 (1 байт до 128 МиБ). Если установлено `null`, ограничение не применяется. По умолчанию: 65536 (64 КиБ). Поведение параметра: - *поддерживается*, если `history` установлено в ”binary” или ”JSON”
time_from	timestamp	Возвращать только значения, полученные после указанного времени или в указанное время.
time_till	timestamp	Возвращать только значения, полученные до указанного времени или в указанное время.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `clock`, `ns`. Не поддерживает сортировку по `ns`, если данные истории хранятся в ClickHouse.

Parameter	Type	Description
search	object	Возвращать результаты, соответствующие заданному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств, а значения — строки для поиска. Если дополнительные параметры не указаны, будет выполнен поиск `LIKE "%…%"`. Не поддерживает поиск по `value`, если `history` установлено в 6 (JSON).
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение данных истории элемента данных

Возвращает 10 последних значений, полученных от числового элемента данных типа float.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23296",
            "clock": "1351090996",
            "value": "0.085",
            "ns": "563157632"
        },
        {
            "itemid": "23296",
            "clock": "1351090936",
            "value": "0.16",
            "ns": "549216402"
```

```
        },
        {
            "itemid": "23296",
            "clock": "1351090876",
            "value": "0.18",
            "ns": "537418114"
        },
        {
            "itemid": "23296",
            "clock": "1351090816",
            "value": "0.21",
            "ns": "522659528"
        },
        {
            "itemid": "23296",
            "clock": "1351090756",
            "value": "0.215",
            "ns": "507809457"
        },
        {
            "itemid": "23296",
            "clock": "1351090696",
            "value": "0.255",
            "ns": "495509699"
        },
        {
            "itemid": "23296",
            "clock": "1351090636",
            "value": "0.36",
            "ns": "477708209"
        },
        {
            "itemid": "23296",
            "clock": "1351090576",
            "value": "0.375",
            "ns": "463251343"
        },
        {
            "itemid": "23296",
            "clock": "1351090516",
            "value": "0.315",
            "ns": "447947017"
        },
        {
            "itemid": "23296",
            "clock": "1351090456",
            "value": "0.275",
            "ns": "435307141"
        }
    ],
    "id": 1
}
```


Источник

CHistory::get() в *ui/include/classes/api/services/CHistory.php*.

history.push

Описание

`object history.push(  /  itemHistoryData)`

Этот метод позволяет отправлять данные истории элементов на сервер Zabbix.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Данные истории элемента данных для отправки.

Метод поддерживает следующие параметры.

Parameter	Type	Description
itemid	ID	ID связанного элемента данных. Поведение параметра: - *required* if `host` and `key` are not set
host	string	Техническое имя узла сети. Поведение параметра: - *required* if `itemid` is not set
key	string	Ключ элемента данных. Поведение параметра: - *required* if `itemid` is not set
value	mixed	Значение элемента данных. Поведение параметра: - *required*
clock	timestamp	Время, когда было получено значение.
ns	integer	Наносекунды, когда было получено значение.

Возвращаемые значения

`(object)` Возвращает результат операции отправки данных.

Примеры

Отправка данных истории элемента данных

Отправьте данные истории элемента данных на сервер Zabbix для элементов данных ”10600”, ”10601” и ”999999”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
```

```
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

См. также

- Элементы данных trapper
- Элементы данных HTTP agent
- Узел сети
- Элемент данных

Источник

CHistory::push() в *ui/include/classes/api/services/CHistory.php*.

Карта значений

Этот класс предназначен для работы с картами значений.

Ссылки на объекты:

- Карта значений
  - Сопоставления значений

Доступные методы:

- valuemap.create - создать новые карты значений
- valuemap.delete - удалить карты значений
- valuemap.get - получить карты значений
- valuemap.update - обновить карты значений

Объект карты значений

Следующие объекты напрямую связаны с API `valuemap`.

Карта значений

Объект карты значений имеет следующие свойства.

Property	Type	Description
valuemapid	ID	ID карты значений. Property behavior: - *read-only* - *required* for update operations
hostid	ID	ID узла сети или шаблона, которому принадлежит карта значений. Property behavior: - *constant* - *required* for create operations
name	string	Имя карты значений. Property behavior: - *required* for create operations
mappings	array	Сопоставления значений для текущей карты значений. Объект сопоставления подробно описан ниже. Property behavior: - *required* for create operations
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных карт значений с уже существующими. Если не указан, генерируется автоматически. Property behavior: - *supported* if the value map belongs to a template

Сопоставления значений

Объект сопоставлений значений определяет сопоставления значений карты значений. Он имеет следующие свойства.

Property	Type	Description
type	integer	Тип сопоставления. Возможные значения: 0 - *(по умолчанию)* сопоставление будет применено, если значение равно; 1 - сопоставление будет применено, если значение больше или равно[1]; 2 - сопоставление будет применено, если значение меньше или равно[1]; 3 - сопоставление будет применено, если значение находится в диапазоне (границы диапазона включены; можно задать несколько диапазонов, разделенных запятой)[1]; 4 - сопоставление будет применено, если значение соответствует регулярному выражению[2]; 5 - если совпадения не найдены, сопоставление не будет применено, и будет использовано значение по умолчанию. Если `type` имеет значение ”0”, ”1”, ”2”, ”3”, ”4”, то `value` не может быть пустым. Если `type` имеет значение ”5”, то `value` должен быть пустым.
value	string	Исходное значение. Property behavior: - *required* если `type` имеет значение ”1”, ”2”, ”3”, ”4” - *supported* если `type` имеет значение ”5”

Property	Type	Description
newvalue	string	Значение, в которое сопоставляется исходное значение. Property behavior: *- required*

[1] поддерживается только для элементов данных с типом значения "numeric unsigned", "numeric float".
 [2] поддерживается только для элементов данных с типом значения "character".

valuemap.create

Описание

`object valuemap.create(object/array valuemaps)`

Этот метод позволяет создавать новые карты значений.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Карты значений для создания.

Метод принимает карты значений со стандартными свойствами карты значений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных карт значений в свойстве `valuemapids`. Порядок возвращаемых идентификаторов соответствует порядку переданных карт значений.

Примеры

Создание карты значений

Создайте одну карту значений с двумя сопоставлениями.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
```

```
            "1"
        ]
    },
    "id": 1
}
```

Источник

CValueMap::create() в *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Описание

`object valuemap.delete(array valuemapids)`

Этот метод позволяет удалять карты значений.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе User roles.

Параметры

( ) ID удаляемых преобразований значений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных преобразований значений, указанных в свойстве `valuemapids`.

Примеры

Удаление нескольких карт значений

Удаление двух карт значений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Источник

CValueMap::delete() в *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Описание

`integer/array valuemap.get(object parameters)`

Метод позволяет получать карты значений в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
valuemapids	ID/array	Возвращать только карты значений с указанными ID.
selectMappings	query	Возвращать сопоставления значений для текущей карты значений в свойстве `mappings`. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `valuemapid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение карт значений

Получить все настроенные карты значений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status"
        },
        {
            "valuemapid": "5",
            "name": "APC Battery Status"
        },
        {
            "valuemapid": "7",
            "name": "Dell Open Manage System Status"
        }
    ],
    "id": 1
}
```


Получить одну карту значений с ее сопоставлениями.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
                },
                {
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
```

```
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
        }
    ],
    "id": 1
}
```


Источник

CValueMap::get() в *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Описание

`object valuemap.update(object/array valuemaps)`

Этот метод позволяет обновлять существующие карты значений.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства карты значений, которые нужно обновить.

Свойство `valuemapid` должно быть определено для каждой карты значений, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных преобразований значений, указанных в свойстве`valuemapids`.

Примеры

Изменение имени карты значений

Измените имя карты значений на ”Device status”.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Изменение сопоставлений для одной карты значений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Источник

CValueMap::update() в *ui/include/classes/api/services/CValueMap.php*.

Карта значков

Этот класс предназначен для работы с картами значков.

Ссылки на объекты:

- Карта значков
- Сопоставление значков

Доступные методы:

- iconmap.create - создать новые карты значков
- iconmap.delete - удалить карты значков
- iconmap.get - получить карты значков
- iconmap.update - обновить карты значков

Объект карты значков

Следующие объекты напрямую связаны с API `iconmap`.

Карта значков

Объект карты значков имеет следующие свойства.

Property	Type	Description
iconmapid	ID	ID карты значков. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
default_iconid	ID	ID значка по умолчанию. Поведение свойства: - *обязательно* для операций создания
name	string	Имя карты значков. Поведение свойства: - *обязательно* для операций создания

Сопоставление значков

Объект сопоставления значков определяет конкретный значок, который будет использоваться для узлов сети с определенным значением поля инвентаризации. Он имеет следующие свойства.

Property	Type	Description
iconid	ID	ID значка, используемого сопоставлением значков. Поведение свойства: - *required*
expression	string	Выражение, с которым сравнивается поле инвентаризации. Поведение свойства: - *required*
inventory_link	integer	ID поля инвентаризации узла сети. См. объект инвентаризации узла сети для списка поддерживаемых полей инвентаризации. Поведение свойства: - *required*
sortorder	integer	Позиция сопоставления значков в карте значков. Поведение свойства: - *read-only*

iconmap.create

Описание

`object iconmap.create(object/array iconMaps)`

Этот метод позволяет создавать новые карты значков.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Карта значков для создания.

В дополнение к стандартным свойствам карты значков, метод принимает следующие параметры.

Параметр	Type	Описание
mappings	array	Сопоставления значков, которые будут созданы для карты значков. Поведение параметра: - *required*

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных соответствий иконок, указанных в свойстве `iconmapids`. Порядок возвращаемых ID совпадает с порядком переданных соответствий иконок.

Примеры

Создание карты значков

Создайте карту значков для отображения узлов сети разных типов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
        "name": "Type icons",
        "default_iconid": "2",
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Сопоставление значков

Источник

CIconMap::create() в *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Описание

`object iconmap.delete(array iconMapIds)`

Этот метод позволяет удалять карты значков.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых соответствий иконок.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных соответствий иконок, указанных в свойстве `iconmapids`.

Примеры

Удаление нескольких карт значков

Удаление двух карт значков.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CIconMap::delete() в *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Описание

`integer/array iconmap.get(object parameters)`

Метод позволяет получать карты значков в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
iconmapids	ID/array	Возвращать только icon maps с указанными ID.
sysmapids	ID/array	Возвращать только icon maps, которые используются в указанных maps.
selectMappings	query	Возвращать свойство `mappings` с используемыми icon mappings.
sortfield	string/array	Сортировать результат по указанным свойствам. Possible values: `iconmapid`, `name`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение карты значков

Получить все данные о карте значков "3".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
```

```
                    "inventory_link": "1",
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

См. также

- Сопоставление значков

Источник

CIconMap::get() в *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Описание

`object iconmap.update(object/array iconMaps)`

Этот метод позволяет обновлять существующие карты значков.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства карты значков, которые нужно обновить.

Для каждой карты значков должно быть определено свойство `iconmapid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств карты значков, метод принимает следующие параметры.

Parameter	Type	Description
mappings	array	Сопоставления значков для замены существующих сопоставлений значков.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных соответствий иконок, указанных в свойстве `iconmapids`.

Примеры

Переименовать карту значков

Переименуйте карту значков в "OS icons".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

См. также

- Сопоставление значков

Источник

CIconMap::update() в *ui/include/classes/api/services/CIconMap.php*.

Каталог пользователей

Этот класс предназначен для работы с каталогами пользователей.

Ссылки на объекты:

- Каталог пользователей
  - Сопоставления типов медиа
  - Сопоставления групп автоматической регистрации

Доступные методы:

- userdirectory.create - создать новый каталог пользователей
- userdirectory.delete - удалить каталог пользователей
- userdirectory.get - получить каталог пользователей
- userdirectory.update - обновить каталог пользователей
- userdirectory.test - проверить подключение к каталогу пользователей

Объект каталога пользователей

Следующие объекты напрямую связаны с API `userdirectory`.

Каталог пользователей

Объект каталога пользователей имеет следующие свойства.

Свойство	Тип	Описание
userdirectoryid	ID	Идентификатор каталога пользователей. Если каталог пользователей удалён, значение свойства `userdirectoryid` объекта пользователя устанавливается равным "0" для всех пользователей, связанных с удалённым каталогом пользователей. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
idp_type	integer	Тип протокола аутентификации, используемого поставщиком удостоверений для каталога пользователей. Обратите внимание, что может существовать только один каталог пользователей типа SAML. Возможные значения: 1 - каталог пользователей типа LDAP; 2 - каталог пользователей типа SAML. Поведение свойства: - *обязательно* для операций создания
group_name	string	Атрибут каталога пользователей LDAP/SAML, содержащий имя группы, используемое для сопоставления групп между каталогом пользователей LDAP/SAML и Zabbix. Пример: *cn* Поведение свойства: - *обязательно*, если `provision_status` имеет значение ”Enabled”, а `saml_jit_status` объекта аутентификации имеет значение ”Enabled for configured SAML IdPs”
user_username	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если `scim_status` имеет значение ”SCIM provisioning is enabled”), содержащий имя пользователя, которое используется как значение свойства `name` объекта пользователя при подготовке пользователя. Примеры: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если `scim_status` имеет значение ”SCIM provisioning is enabled”), содержащий фамилию пользователя, которая используется как значение свойства `surname` объекта пользователя при подготовке пользователя. Примеры: *sn*, *surname*, *lastName*
provision_status	integer	Состояние подготовки каталога пользователей. Возможные значения: 0 - *(по умолчанию)* отключено (подготовка пользователей, созданных этим каталогом пользователей, отключена); 1 - включено (подготовка пользователей, созданных этим каталогом пользователей, включена; кроме того, подготовка LDAP или SAML (`ldap_jit_status` или `saml_jit_status` объекта аутентификации) должна быть включена).
provision_groups	array	Массив объектов сопоставлений групп при подготовке для сопоставления шаблона группы пользователей LDAP/SAML с группой пользователей и ролью пользователя Zabbix. Поведение свойства: - *обязательно*, если `provision_status` имеет значение ”Enabled”
provision_media	array	Массив объектов сопоставлений типов средств связи для сопоставления атрибутов средств связи пользователя LDAP/SAML (например, электронной почты) со средствами связи пользователя Zabbix для отправки уведомлений.
Свойства, специфичные для LDAP:		
name	string	Уникальное имя каталога пользователей. Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type LDAP”

Свойство	Тип	Описание
host	string	Имя узла сети, IP-адрес или URI сервера LDAP. URI должен содержать схему (`ldap://` или `ldaps://`), узел сети и порт (необязательно). Примеры: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type LDAP”
port	integer	Порт сервера LDAP. Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type LDAP”
base_dn	string	Базовый путь каталога пользователей LDAP к учётным записям пользователей. Примеры: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (для OpenLDAP) *DC=company,DC=com* (для Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (для прямой привязки пользователя; заполнитель ”*%{user}*” является обязательным) Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type LDAP”
search_attribute	string	Атрибут каталога пользователей LDAP, по которому определяется учётная запись пользователя на основе информации, предоставленной в запросе входа. Примеры: *uid* (для OpenLDAP) *sAMAccountName* (для Microsoft Active Directory) Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type LDAP”
bind_dn	string	Учётная запись сервера LDAP для привязки и поиска на сервере LDAP. Для прямой привязки пользователя и анонимной привязки `bind_dn` должен быть пустым. Примеры: *uid=ldap_search,ou=system* (для OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (для Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type LDAP”

Свойство	Тип	Описание
bind_password	string	Пароль учётной записи LDAP для привязки и поиска на сервере LDAP. Для прямой привязки пользователя и анонимной привязки `bind_password` должен быть пустым. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type LDAP”
description	string	Описание каталога пользователей. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type LDAP”
group_basedn	string	Базовый путь каталога пользователей LDAP к группам; используется для настройки проверки членства пользователя в каталоге пользователей LDAP. Игнорируется при подготовке пользователя, если задано `group_membership`. Пример: *ou=Groups,dc=example,dc=com* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type LDAP”
group_filter	string	Строка фильтра для получения групп каталога пользователей LDAP, членом которых является пользователь; используется для настройки проверки членства пользователя в каталоге пользователей LDAP. Игнорируется при подготовке пользователя, если задано `group_membership`. Поддерживаемые заполнители `group_filter`: *%{attr}* - атрибут поиска (заменяется значением свойства `search_attribute`); *%{groupattr}* - атрибут группы (заменяется значением свойства `group_member`); *%{host}* - имя узла сети, IP-адрес или URI сервера LDAP (заменяется значением свойства `host`); *%{user}* - имя пользователя Zabbix. По умолчанию: *(%{groupattr}=%{user})* Примеры: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* сопоставит ”User1”, если объект группы LDAP содержит атрибут ”*member*” со значением ”*uid=User1,ou=Users,dc=example,dc=com*”, и вернёт группу, членом которой является ”User1”; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* сопоставит ”User1”, если объект группы LDAP содержит атрибут, указанный в свойстве `group_member`, со значением ”*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*”, и вернёт группу, членом которой является ”User1”. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type LDAP”

Свойство	Тип	Описание
group_member	string	Атрибут каталога пользователей LDAP, содержащий сведения об участниках группы; используется для настройки проверки членства пользователя в каталоге пользователей LDAP. Игнорируется при подготовке пользователя, если задано `group_membership`. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type LDAP"
group_membership	string	Атрибут каталога пользователей LDAP, содержащий сведения о группах, в которые входит пользователь. Пример: *memberOf* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type LDAP"
search_filter	string	Пользовательская строка фильтра, используемая для поиска и аутентификации пользователя в каталоге пользователей LDAP на основе информации, предоставленной в запросе входа. Поддерживаемые заполнители `search_filter`: *%{attr}* - имя атрибута поиска (например, *uid*, *sAMAccountName*); *%{user}* - имя пользователя Zabbix. По умолчанию: *(%{attr}=%{user})* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type LDAP"
start_tls	integer	Параметр конфигурации сервера LDAP, позволяющий защитить обмен данными с сервером LDAP с помощью Transport Layer Security (TLS). Обратите внимание, что для узлов сети, использующих протокол `ldaps://`, параметр `start_tls` должен иметь значение "Disabled". Возможные значения: 0 - *(по умолчанию)* отключено; 1 - включено. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type LDAP"
user_ref_attr	string	Атрибут каталога пользователей LDAP, используемый для ссылки на объект пользователя. Значение `user_ref_attr` используется для получения значений из указанного атрибута в каталоге пользователей и подстановки их вместо заполнителя *%{ref}* в строке `group_filter`. Примеры: *cn*, *uid*, *member*, *uniqueMember* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type LDAP"
Свойства, специфичные для SAML:		

Свойство	Тип	Описание
idp_entityid	string	URI, идентифицирующий поставщика удостоверений и используемый для обмена сообщениями SAML с поставщиком удостоверений. Пример: *https://idp.example.com/idp* Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type SAML”
sp_entityid	string	URL или любая строка, идентифицирующая поставщика услуг поставщика удостоверений. Примеры: *https://idp.example.com/sp* *zabbix* Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type SAML”
username_attribute	string	Атрибут каталога пользователей SAML (а также атрибут SCIM, если `scim_status` имеет значение ”SCIM provisioning is enabled”), содержащий имя пользователя, которое при аутентификации сравнивается со значением свойства `username` объекта пользователя. Примеры: *uid*, *userprincipalname*, *samaccountname*, *username*, *userusername*, *urn:oid:0.9.2342.19200300.100.1.1*, *urn:oid:1.3.6.1.4.1.5923.1.1.1.13*, *urn:oid:0.9.2342.19200300.100.1.44* Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type SAML”
sso_url	string	URL службы единого входа SAML поставщика удостоверений, на который Zabbix отправляет запросы аутентификации SAML. Пример: *http://idp.example.com/idp/sso/saml* Поведение свойства: - *обязательно*, если `idp_type` имеет значение ”User directory of type SAML”
slo_url	string	URL службы единого выхода SAML поставщика удостоверений, на который Zabbix отправляет запросы выхода из системы SAML. Пример: *https://idp.example.com/idp/slo/saml* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type SAML”
encrypt_nameid	integer	Указывает, следует ли шифровать идентификатор имени SAML. Возможные значения: 0 - *(по умолчанию)* не шифровать идентификатор имени; 1 - шифровать идентификатор имени. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение ”User directory of type SAML”

Свойство	Тип	Описание
encrypt_assertions	integer	Указывает, следует ли шифровать утверждения SAML. Возможные значения: 0 - *(по умолчанию)* не шифровать утверждения; 1 - шифровать утверждения. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
nameid_format	string	Формат идентификатора имени поставщика услуг поставщика удостоверений SAML. Примеры: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
scim_status	integer	Указывает, включена или отключена подготовка SCIM для SAML. Возможные значения: 0 - *(по умолчанию)* подготовка SCIM отключена; 1 - подготовка SCIM включена. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
sign_assertions	integer	Указывает, следует ли подписывать утверждения SAML подписью SAML. Возможные значения: 0 - не подписывать утверждения; 1 - *(по умолчанию)* подписывать утверждения. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
sign_authn_requests	integer	Указывает, следует ли подписывать запросы AuthN SAML подписью SAML. Возможные значения: 0 - не подписывать запросы AuthN; 1 - *(по умолчанию)* подписывать запросы AuthN. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
sign_messages	integer	Указывает, следует ли подписывать сообщения SAML подписью SAML. Возможные значения: 0 - не подписывать сообщения; 1 - *(по умолчанию)* подписывать сообщения. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"

Свойство	Тип	Описание
sign_logout_requests	integer	Указывает, следует ли подписывать запросы выхода из системы SAML подписью SAML. Возможные значения: 0 - не подписывать запросы выхода из системы; 1 - *(по умолчанию)* подписывать запросы выхода из системы. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
sign_logout_responses	integer	Указывает, следует ли подписывать ответы выхода из системы SAML подписью SAML. Возможные значения: 0 - не подписывать ответы выхода из системы; 1 - *(по умолчанию)* подписывать ответы выхода из системы. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML"
idp_certificate	string	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа (SSO) SAML. Примеры: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для записи*
sp_private_key	string	Содержимое закрытого ключа поставщика услуг (SP) для настройки службы единого входа (SSO) SAML. Обеспечивает безопасную аутентификацию и обмен данными с поставщиком удостоверений (IdP). Примеры: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для записи*
sp_certificate	string	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа (SSO) SAML. Примеры: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для записи*

Свойство	Тип	Описание
idp_certificate_hash	string	Хеш MD5 значения idp_certificate. Для пустого idp_certificate возвращается пустая строка. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для чтения*
sp_private_key_hash	string	Хеш MD5 значения sp_private_key. Для пустого sp_private_key возвращается пустая строка. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для чтения*
sp_certificate_hash	string	Хеш MD5 значения sp_certificate. Для пустого sp_certificate возвращается пустая строка. Поведение свойства: - *поддерживается*, если `idp_type` имеет значение "User directory of type SAML", а `$SSO['CERT_STORAGE']` имеет значение `database` в файле конфигурации веб-интерфейса - *только для чтения*

Сопоставления типов медиа

Объект сопоставлений типов медиа имеет следующие свойства.

Property	Type	Description
userdirectory_mediaid	ID	ID сопоставления типа медиа. Поведение свойства: - *только для чтения*
name	string	Отображаемое имя в списке сопоставлений типов медиа. Поведение свойства: - *обязательно*
mediatypeid	ID	ID типа медиа, который будет создан; используется как значение свойства объекта Media `mediatypeid`. Поведение свойства: - *обязательно*
attribute	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если для `scim_status` установлено значение "SCIM provisioning is enabled"), который содержит медиа пользователя (например, *user@example.com*), используемый как значение свойства объекта Media `sendto`. Если он присутствует в данных, полученных от поставщика удостоверений LDAP/SAML, и его значение не пустое, это вызовет создание медиа для подготовленного пользователя. Поведение свойства: - *обязательно*
active	integer	Значение свойства `active` медиа пользователя при создании медиа для подготовленного пользователя. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.

Property	Type	Description
severity	integer	Значение свойства `severity` медиа пользователя при создании медиа для подготовленного пользователя. По умолчанию: 63.
period	string	Значение свойства `period` медиа пользователя при создании медиа для подготовленного пользователя. По умолчанию: 1-7,00:00-24:00.

Сопоставления групп при автоматическом создании пользователей

Сопоставления групп при автоматическом создании пользователей имеют следующие свойства.

Property	Type	Description
name	string	Полное имя группы (например, *Zabbix administrators*) в каталоге пользователей LDAP/SAML (а также SCIM, если `scim_status` имеет значение "SCIM provisioning is enabled"). Поддерживает символ подстановки "*". Уникально среди всех сопоставлений групп при автоматическом создании пользователей. Property behavior: - *required*
roleid	ID	ID роли пользователя, назначаемой пользователю. Если совпадает несколько сопоставлений групп при автоматическом создании пользователей, пользователю назначается роль с наивысшим типом пользователя (*User*, *Admin* или *Super admin*). Если есть несколько ролей с одним и тем же типом пользователя, пользователю назначается первая роль (отсортированная в алфавитном порядке). Property behavior: - *required*
user_groups	array	Массив объектов ID групп пользователей Zabbix. Каждый объект имеет следующие свойства: `usrgrpid` - `(ID)` ID группы пользователей Zabbix, назначаемой пользователю. Если совпадает несколько сопоставлений групп при автоматическом создании пользователей, пользователю назначаются группы пользователей Zabbix из всех совпавших сопоставлений. Property behavior: - *required*

userdirectory.create

Описание

`object userdirectory.create(object/array userDirectory)`

Этот метод позволяет создавать новые каталоги пользователей.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*.

Параметры

`(object/array)` Каталоги пользователей для создания.

Метод принимает каталоги пользователей со стандартными свойствами каталога пользователей.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы созданных каталогов пользователей в свойстве userdirectoryids. Порядок возвращаемых идентификаторов соответствует порядку переданных каталогов пользователей.

Примеры

Создание каталога пользователей

Создайте каталог пользователей для аутентификации пользователей с помощью StartTLS через LDAP. Обратите внимание, что для аутентификации пользователей через LDAP должна быть включена LDAP-аутентификация.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
        "idp_type": "1",
        "name": "LDAP API server #1",
        "host": "ldap://local.ldap",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "ldapsecretpassword",
        "search_attribute": "uid",
        "start_tls": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Создание каталога пользователей (JIT provisioning включен)

Создайте каталог пользователей для аутентификации пользователей через LDAP (с включенным JIT provisioning). Обратите внимание, что для аутентификации пользователей через LDAP должна быть включена LDAP-аутентификация.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
```

```
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Исходный код

CUserDirectory::create() в *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Описание

`object userdirectory.delete(array userDirectoryIds)`

Этот метод позволяет удалять каталоги пользователей. Каталог пользователей нельзя удалить, если он напрямую используется хотя бы одной группой пользователей.
 Каталог пользователей LDAP по умолчанию нельзя удалить, если `authentication.ldap_configured` установлено в `1` или если после удаления останется меньше каталогов пользователей.

Note:
Этот метод доступен только для типа пользователя *Super admin*.

Параметры

`(array)` Идентификаторы пользовательских каталогов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных каталогов пользователей в свойстве `userdirectoryids`.

Примеры

Удаление нескольких каталогов пользователей

Удаление двух каталогов пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Исходный код

CUserDirectory::delete() в *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Описание

`integer/array userdirectory.get(object parameters)`

Метод позволяет получить каталоги пользователей в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типов пользователей *Super admin*.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
userdirectoryids	ID/array	Возвращать только каталоги пользователей с указанными ID.
selectUsrgrps	query	Возвращать свойство `usrgrps` с группами пользователей, связанными с каталогом пользователей. Поддерживает `count`.

Parameter	Type	Description
selectProvisionMedia	query	Возвращать свойство `provision_media` с сопоставлениями типов медиа, связанными с каталогом пользователей.
selectProvisionGroups	query	Возвращать свойство `provision_groups` с сопоставлениями групп для автоматического создания, связанными с каталогом пользователей.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `name`.
filter	object	Возвращать только те результаты, которые точно соответствуют заданному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений. Поддерживаемые свойства: `userdirectoryid`, `idp_type`, `provision_status`.
search	object	Возвращать результаты, соответствующие заданному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств, а значения — строки для поиска. Если дополнительные параметры не заданы, будет выполнен поиск `LIKE "%…%"`. Поддерживаемые свойства: `name`, `description`. Каталог пользователей типа SAML будет иметь пустое значение свойств `name` и `description`. Оба свойства можно изменить с помощью операции `userdirectory.update`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов
- количество полученных объектов, если использовался параметр `countOutput`

Примеры

Получение каталогов пользователей

Получите все каталоги пользователей с дополнительными свойствами, которые отображают сопоставления типов медиа и сопоставления групп автоматического создания, связанные с каждым каталогом пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
            "name": "",
            "provision_status": "1",
            "description": "",
            "group_name": "groups",
            "user_username": "",
            "user_lastname": "",
            "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
            "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
            "slo_url": "",
            "username_attribute": "uid",
            "sp_entityid": "zabbix",
            "nameid_format": "",
            "sign_messages": "0",
            "sign_assertions": "0",
            "sign_authn_requests": "0",
            "sign_logout_requests": "0",
            "sign_logout_responses": "0",
            "encrypt_nameid": "0",
            "encrypt_assertions": "0",
            "scim_status": "1",
            "provision_media": [
                {
                    "userdirectory_mediaid": "1",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "1",
                    "user_groups": [
                        {
                            "usrgrpid": "13"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "2",
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
```

```
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "userdirectory_mediaid": "2",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
```

```
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
    "id": 1
}
```

См. также

- Группа пользователей

Источник

CUserDirectory::get() в *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Описание

`object userdirectory.test(array userDirectory)`

Этот метод позволяет проверить настройки подключения к каталогу пользователей.

Note:
Этот метод также позволяет проверить, какие настроенные данные соответствуют настройкам каталога пользователей для подготовки пользователей (например, какая роль пользователя, какие группы пользователей, какие пользовательские медиа будут назначены пользователю). Для такого типа проверки API-запрос следует выполнять для каталога пользователей, у которого `provision_status` установлен в enabled.

Note:
Этот метод доступен только для типа пользователя *Super admin*.

Параметры

`(object)` Свойства каталога пользователей.

Поскольку API `userdirectory.get` не возвращает поле `bind_password`, следует указать `userdirectoryid` и/или `bind_password`.
 Помимо стандартных свойств каталога пользователей, метод принимает следующие параметры.

Parameter	Type	Description
test_username	string	Имя пользователя для проверки в каталоге пользователей.
test_password	string	Пароль, связанный с именем пользователя, для проверки в каталоге пользователей.

Возвращаемые значения

`(bool)` Возвращает true в случае успеха.

Примеры

Тестирование каталога пользователей для существующего пользователя

Проверка каталога пользователей ”3” для ”user1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
```

```
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "password",
        "test_username": "user1",
        "test_password": "password"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Проверка каталога пользователей для несуществующего пользователя

Проверка каталога пользователей ”3” для несуществующего ”user2”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "test_username": "user2",
        "test_password": "password"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

Тестирование каталога пользователей для подготовки пользователей

Проверьте каталог пользователей ”3”, чтобы определить, какие настроенные данные соответствуют параметрам каталога пользователей для подготовки ”user3” (например, какая роль пользователя, группы пользователей и средства оповещения будут назначены пользователю).

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "2",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
```

```
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "test_username": "user3",
        "test_password": "password"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Источник

CUserDirectory::test() в *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Описание

`object userdirectory.update(object/array userDirectory)`

Этот метод позволяет обновлять существующие каталоги пользователей.

Note:
Этот метод доступен только для пользователей типа *Super admin*.

Параметры

`(object/array)` Свойства каталога пользователей, которые необходимо обновить.

Свойство `userdirectoryid` должно быть определено для каждого каталога пользователей, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных каталогов пользователей в свойстве `userdirectoryids`.

Примеры

Обновление bind-пароля для каталога пользователей

Установите новый bind-пароль для каталога пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
```

```
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Обновление сопоставлений для каталога пользователей

Обновите сопоставления групп provisioning и сопоставления типов оповещений для каталога пользователей "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "2",
        "provision_media": [
                {
                    "userdirectory_mediaid": "2"
                }
            ],
            "provision_groups": [
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        },
                        {
                            "usrgrpid": "11"
                        }
                    ]
                }
            ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
```

```
}
```

Источник

CUserDirectory::update() в *ui/include/classes/api/services/CUserDirectory.php*.

Конфигурация

Этот класс предназначен для экспорта и импорта данных конфигурации Zabbix.

Доступные методы:

- configuration.export - экспорт конфигурации
- configuration.import - импорт конфигурации
- configuration.importcompare - сравнение файла импорта с текущими элементами системы

configuration.export

Описание

`string configuration.export(object parameters)`

Этот метод позволяет экспортировать данные конфигурации в виде сериализованной строки.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие объекты для экспорта и используемый формат.

Parameter	Type	Description
format	string	Формат, в котором должны быть экспортированы данные. Возможные значения: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - необработанный массив PHP. Поведение параметра: *- required*
prettyprint	boolean	Сделать вывод более удобочитаемым для человека, добавив отступы. Возможные значения: `true` - добавить отступы; `false` - *(по умолчанию)* не добавлять отступы.

Parameter	Type	Description
options	object	Объекты, которые нужно экспортировать. Объект `options` содержит следующие параметры: `host_groups` - `(array)` идентификаторы групп узлов сети для экспорта; `hosts` - `(array)` идентификаторы узлов сети для экспорта; `images` - `(array)` идентификаторы изображений для экспорта; `maps` - `(array)` идентификаторы карт для экспорта; `mediaTypes` - `(array)` идентификаторы типов медиа для экспорта; `template_groups` - `(array)` идентификаторы групп шаблонов для экспорта; `templates` - `(array)` идентификаторы шаблонов для экспорта; `dashboards` - `(array)` идентификаторы панелей мониторинга для экспорта; `global_regexes` - `(array)` идентификаторы глобальных регулярных выражений для экспорта. Пользователи типов *Admin* и *User* могут экспортировать только те объекты, для которых у них есть права *read-only* или *read-write* permission, а также изображения, но не типы медиа. Поведение параметра: - *required*.

Возвращаемые значения

`(string)` Возвращает сериализованную строку, содержащую запрошенные данные конфигурации.

Примеры

Экспорт шаблона

Экспортируйте конфигурацию шаблона ”10571” в виде XML-строки.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```

Источник

CConfiguration::export() в *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Описание

`boolean configuration.import(object parameters)`

Этот метод позволяет импортировать данные конфигурации из сериализованной строки.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие данные для импорта и правила обработки этих данных.

Parameter	Type	Description
format	string	Формат сериализованной строки. Возможные значения: `yaml` - YAML; `xml` - XML; `json` - JSON. Поведение параметра: - *required*
source	string	Сериализованная строка, содержащая данные конфигурации. Поведение параметра: - *required*
rules	object	Правила импорта новых и существующих объектов. Пользователи типа *Admin* могут импортировать только те объекты, для которых у них есть разрешение на *read-write*, а также карты. Например, узел сети и его сущности (элементы данных, триггеры, графики и т. д.) могут быть импортированы только в том случае, если группа пользователей имеет разрешение на группу узлов сети, к которой будет принадлежать импортируемый узел сети. Изображения и типы оповещений не могут быть импортированы пользователями типа *Admin*. Параметр `rules` подробно описан в таблице ниже. Поведение параметра: - *required*
returnMissingObjects	boolean	Правила, определяющие, должен ли объект ответа содержать информацию об отсутствующих объектах; по умолчанию: `false`.

Attention:
Если правила не заданы, конфигурация не будет обновлена.

Объект `rules` поддерживает следующие параметры.

Parameter	Type	Description
dashboards	object	Правила импорта панелей. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые панели; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true` и в системе присутствует конкретная импортируемая панель, она будет полностью обновлена. Страницы и виджеты будут полностью заменены новыми; по умолчанию: `false`.
discoveryRules	object	Правила импорта правил LLD. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые правила LLD; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие правила LLD будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, правила LLD, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
global_regexes	object	Правила импорта глобальных регулярных выражений. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые глобальные регулярные выражения; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие глобальные регулярные выражения будут обновлены, при этом их подвыражения будут полностью перезаписаны; по умолчанию: `false`.
graphs	object	Правила импорта графиков. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые графики; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие графики будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, графики, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
host_groups	object	Правила импорта групп узлов сети. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые группы узлов сети; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие группы узлов сети будут обновлены; по умолчанию: `false`.
hosts	object	Правила импорта узлов сети. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые узлы сети; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие узлы сети будут обновлены; по умолчанию: `false`.

Parameter	Type	Description
httptests	object	Правила импорта веб-сценариев. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые веб-сценарии; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие веб-сценарии будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено в `true`, веб-сценарии, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
images	object	Правила импорта изображений. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые изображения; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие изображения будут обновлены; по умолчанию: `false`.
items	object	Правила импорта элементов данных. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые элементы данных; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие элементы данных будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено в `true`, элементы данных, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
maps	object	Правила импорта карт. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые карты; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие карты будут обновлены; по умолчанию: `false`.
mediaTypes	object	Правила импорта типов оповещений. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые типы оповещений; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие типы оповещений будут обновлены; по умолчанию: `false`.
templateDashboards	object	Правила импорта панелей шаблонов. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено в `true`, будут созданы новые панели шаблонов; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено в `true`, существующие панели шаблонов будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено в `true`, панели шаблонов, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.

Parameter	Type	Description
template_groups	object	Правила импорта групп шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые группы шаблонов; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие группы шаблонов будут обновлены; по умолчанию: `false`.
templateLinkage	object	Правила импорта связей шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, шаблоны, которые не связаны с импортируемым узлом сети или шаблоном, но присутствуют в импортируемых данных, будут связаны; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, шаблоны, которые связаны с импортируемым узлом сети или шаблоном, но отсутствуют в импортируемых данных, будут отвязаны без удаления сущностей (элементов данных, триггеров и т. д.), унаследованных от отвязанных шаблонов; по умолчанию: `false`.
templates	object	Правила импорта шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые шаблоны; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие шаблоны будут обновлены; по умолчанию: `false`.
triggers	object	Правила импорта триггеров. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые триггеры; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие триггеры будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, триггеры, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
valueMaps	object	Правила импорта карт значений узла сети или шаблона. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые карты значений; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие карты значений будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, карты значений, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.

Возвращаемые значения

( ) Возвращает `true` при успешном импорте.

Примеры

Импорт шаблона

Импортируйте конфигурацию шаблона, содержащуюся в XML-строке. Если какие-либо элементы данных или триггеры в XML-строке отсутствуют, они будут удалены из базы данных, а все остальное останется без изменений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.import",
    "params": {
        "format": "xml",
        "rules": {
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": false
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


Источник

CConfiguration::import() в *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

Описание

`array configuration.importcompare(object parameters)`

Этот метод позволяет сравнить файл импорта с текущими элементами системы и показывает, что будет изменено, если этот файл импорта будет импортирован.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие возможные данные для импорта и правила обработки этих данных.

Parameter	Type	Description
format	string	Формат сериализованной строки. Возможные значения: `yaml` - YAML; `xml` - XML; `json` - JSON. Поведение параметра: - *required*
source	string	Сериализованная строка, содержащая данные конфигурации. Поведение параметра: - *required*
rules	object	Правила сравнения новых и существующих объектов. Пользователи типов *Admin* и *User* могут сравнивать новые объекты только с существующими, для которых у них есть права *read-only* или *read-write* permission. Параметр `rules` подробно описан в таблице ниже. Поведение параметра: - *required*
returnMissingObjects	boolean	Правила, указывающие, должен ли объект ответа содержать информацию об отсутствующих объектах; по умолчанию: `false`.

Attention:
Если правила не заданы, обновлять будет нечего, и результат будет пустым.

Note:
Сравнение будет выполняться только для групп узлов сети и шаблонов. Триггеры и графики будут сравниваться только для импортируемых шаблонов, все остальные будут считаться ”новыми”.

Объект `rules` поддерживает следующие параметры.

Parameter	Type	Description
dashboards	object	Правила импорта панелей мониторинга. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые панели мониторинга будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true` и определенная импортируемая панель мониторинга присутствует в системе, она будет полностью обновлена. Страницы и виджеты будут полностью заменены новыми; по умолчанию: `false`.

Parameter	Type	Description
discoveryRules	object	Правила импорта правил LLD. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые правила LLD будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие правила LLD будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, правила LLD, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
global_regexes	object	Правила импорта глобальных регулярных выражений. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые глобальные регулярные выражения будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие глобальные регулярные выражения будут обновлены, при этом их подвыражения будут полностью перезаписаны; по умолчанию: `false`.
graphs	object	Правила импорта графиков. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые графики будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие графики будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, графики, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
host_groups	object	Правила импорта групп узлов сети. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые группы узлов сети будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие группы узлов сети будут обновлены; по умолчанию: `false`.

Parameter	Type	Description
httptests	object	Правила импорта веб-сценариев. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые веб-сценарии будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие веб-сценарии будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, веб-сценарии, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
images	object	Правила импорта изображений. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые изображения будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие изображения будут обновлены; по умолчанию: `false`. Этот параметр не влияет на результат. Он разрешен только для согласованности с `configuration.import`.
items	object	Правила импорта элементов данных. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые элементы данных будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие элементы данных будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, элементы данных, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.

Parameter	Type	Description
maps	object	Правила импорта карт. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые карты будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие карты будут обновлены; по умолчанию: `false`. Этот параметр не влияет на результат. Он разрешен только для согласованности с `configuration.import`.
mediaTypes	object	Правила импорта типов медиа. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые типы медиа будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие типы медиа будут обновлены; по умолчанию: `false`. Этот параметр не влияет на результат. Он разрешен только для согласованности с `configuration.import`.
templateDashboards	object	Правила импорта панелей мониторинга шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые панели мониторинга шаблонов будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие панели мониторинга шаблонов будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, панели мониторинга шаблонов, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
template_groups	object	Правила импорта групп шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые группы шаблонов будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие группы шаблонов будут обновлены; по умолчанию: `false`.

Parameter	Type	Description
templateLinkage	object	Правила импорта связей шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, шаблоны, которые не связаны с импортируемым узлом сети или шаблоном, но присутствуют в импортируемых данных, будут связаны; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, шаблоны, которые связаны с импортируемым узлом сети или шаблоном, но отсутствуют в импортируемых данных, будут отвязаны без удаления унаследованных от отвязанных шаблонов сущностей (элементов данных, триггеров и т. д.); по умолчанию: `false`.
templates	object	Правила импорта шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые шаблоны будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие шаблоны будут обновлены; по умолчанию: `false`.
triggers	object	Правила импорта триггеров. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые триггеры будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие триггеры будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, триггеры, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.

Parameter	Type	Description
valueMaps	object	Правила импорта карт значений узла сети или шаблона. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, новые карты значений будут созданы; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие карты значений будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, карты значений, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.

Возвращаемые значения

(`array`) Возвращает массив с изменениями в конфигурации, которые будут сделаны.

Примеры

Сравнение импорта шаблона

Сравните шаблон, содержащийся в XML-строке, с текущими элементами системы и покажите, что будет изменено, если этот шаблон будет импортирован.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.importcompare",
    "params": {
        "format": "xml",
        "rules": {
            "discoveryRules": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "graphs": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "host_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "template_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "httptests": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
```

```
                "deleteMissing": true
            },
            "templateLinkage": {
                "createMissing": true,
                "deleteMissing": true
            },
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "templateDashboards": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "templates": {
            "updated": [
                {
                    "before": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "after": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "items": {
                        "added": [
                            {
                                "after": {
```

```
                                "uuid": "648006da5971424ead0c47ddbbf1ea2e",
                                "name": "CPU utilization",
                                "key": "system.cpu.util",
                                "value_type": "FLOAT",
                                "units": "%"
                            },
                            "triggers": {
                                "added": [
                                    {
                                        "after": {
                                            "uuid": "736225012c534ec480c2a66a91322ce0",
                                            "expression": "avg(/New template/system.cpu.util,3m)>70",
                                            "name": "CPU utilization too high on 'New host' for 3 minu
                                            "priority": "WARNING"
                                        }
                                    }
                                ]
                            }
                        }
                    ],
                    "removed": [
                        {
                            "before": {
                                "uuid": "6805d4c39a624a8bab2cc8ab63df1ab3",
                                "name": "CPU load",
                                "key": "system.cpu.load",
                                "value_type": "FLOAT"
                            },
                            "triggers": {
                                "removed": [
                                    {
                                        "before": {
                                            "uuid": "ab4c2526c2bc42e48a633082255ebcb3",
                                            "expression": "avg(/New template/system.cpu.load,3m)>2",
                                            "name": "CPU load too high on 'New host' for 3 minutes",
                                            "priority": "WARNING"
                                        }
                                    }
                                ]
                            }
                        }
                    ],
                    "updated": [
                        {
                            "before": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping"
                            },
                            "after": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping",
                                "delay": "3m"
                            }
                        }
                    ]
                }
            }
        ]
    }
},
```

```
    "id": 1
}
```

Источник

CConfiguration::importcompare() в *ui/include/classes/api/services/CConfiguration.php*.

Корреляция

Этот класс предназначен для работы с корреляциями.

Ссылки на объекты:

- Correlation
- Correlation operation
- Correlation filter
  - Correlation filter condition

Доступные методы:

- correlation.create - создать новые корреляции
- correlation.delete - удалить корреляции
- correlation.get - получить корреляции
- correlation.update - обновить корреляции

Объект корреляции

Следующие объекты напрямую связаны с API корреляций (`correlation`).

Корреляция

Объект корреляции имеет следующие свойства.

Property	Type	Description
correlationid	ID	ID корреляции. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя корреляции. Поведение свойства: - *обязательно* для операций создания
description	string	Описание корреляции.
status	integer	Включена корреляция или отключена. Возможные значения: 0 - *(по умолчанию)* включена; 1 - отключена.

Операция корреляции

Объект операции корреляции определяет операцию, которая будет выполнена при запуске корреляции. Он имеет следующие свойства.

Property	Type	Description
type	integer	Тип операции. Возможные значения: 0 - закрыть старые события; 1 - закрыть новое событие. Поведение свойства: - *required*

Фильтр корреляции

Объект фильтра корреляции определяет набор условий, которые должны быть выполнены для выполнения настроенных операций корреляции. Он имеет следующие свойства.

Property	Type	Description
conditions	array	Набор условий фильтра, используемых для фильтрации результатов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *required*
evaltype	integer	Метод оценки условий фильтра. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Поведение свойства: - *required*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для оценки условий фильтра. Выражение содержит ID, которые ссылаются на конкретные условия фильтра по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *read-only*
formula	string	Определенное пользователем выражение, которое будет использоваться для оценки условий фильтров с пользовательским выражением. Выражение должно содержать ID, которые ссылаются на конкретные условия фильтра по их `formulaid`. ID, используемые в выражении, должны точно совпадать с теми, что определены в условиях фильтра: ни одно условие не может остаться неиспользованным или пропущенным. Поведение свойства: - *required* if `evaltype` is set to ”custom expression”

Условие фильтра корреляции

Объект условия фильтра корреляции определяет конкретное условие, которое должно быть проверено перед выполнением операций корреляции.

Property	Type	Description
type	integer	Тип условия. Возможные значения: 0 - старый тег события; 1 - новый тег события; 2 - новая группа узлов сети события; 3 - пара тегов события; 4 - значение старого тега события; 5 - значение нового тега события. Поведение свойства: - *required*
tag	string	Тег события (старый или новый). Поведение свойства: - *required* если `type` установлено в ”old event tag”, ”new event tag”, ”old event tag value” или ”new event tag value”
groupid	ID	ID группы узлов сети. Поведение свойства: - *required* если `type` установлено в ”new event host group”
oldtag	string	Старый тег события. Поведение свойства: - *required* если `type` установлено в ”event tag pair”
newtag	string	Старый тег события. Поведение свойства: - *required* если `type` установлено в ”event tag pair”
value	string	Значение тега события (старого или нового). Поведение свойства: - *required* если `type` установлено в ”old event tag value” или ”new event tag value”
formulaid	string	Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть задан пользователем при изменении условий фильтра, но будет сгенерирован заново при последующем запросе.
operator	integer	Оператор условия. Поведение свойства: - *required* если `type` установлено в ”new event host group”, ”old event tag value” или ”new event tag value”

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, см. примеры на страницах методов correlation.get и correlation.create.

Для каждого типа условия поддерживаются следующие операторы и значения.

Condition	Condition name	Supported operators	Expected value
2	Группа узлов сети	=, <>	ID группы узлов сети.
4	Значение старого тега события	=, <>, like, not like	string
5	Значение нового тега события	=, <>, like, not like	string

correlation.create

Описание

`object correlation.create(object/array correlations)`

Этот метод позволяет создавать новые корреляции.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Корреляции для создания.

В дополнение к стандартным свойствам корреляции, метод принимает следующие параметры.

Parameter	Type	Description
operations	array	Операции корреляции для создания корреляции. Поведение параметра: - *required*
filter	object	Объект фильтра корреляции для корреляции. Поведение параметра: - *required*

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных корреляций под свойством `correlationids`. Порядок возвращаемых ID совпадает с порядком переданных корреляций.

Примеры

Создание новой корреляции тегов событий

Создайте корреляцию, используя метод оценки `AND/OR`, с одним условием и одной операцией. По умолчанию корреляция будет включена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new event tag correlation",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "type": 1,
                    "tag": "ok"
                }
            ]
        },
        "operations": [
            {
                "type": 0
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Использование пользовательского фильтра выражения

Создайте корреляцию, которая будет использовать пользовательское условие фильтра. Идентификаторы формулы "A" или "B" выбраны произвольно. Тип условия будет "Группа узлов сети" с оператором "<>".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new host group correlation",
        "description": "a custom description",
        "status": 0,
        "filter": {
            "evaltype": 3,
            "formula": "A or B",
            "conditions": [
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "A"
                },
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "B"
                }
            ]
        },
        "operations": [
            {
                "type": 1
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Фильтр корреляции
- Операция корреляции

Источник

CCorrelation::create() в *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Описание

`object correlation.delete(array correlationids)`

Этот метод позволяет удалять корреляции.

> **Note:**
> Этот метод доступен только пользователю типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых корреляций.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных корреляций, переданных в свойстве `correlationids`.

Пример

Удаление нескольких корреляций

Удаление двух корреляций.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Источник

CCorrelation::delete() в *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

Описание

`integer/array correlation.get(object parameters)`

Метод позволяет получить корреляции в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object) Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
correlationids	ID/array	Возвращать только корреляции с указанными ID.
selectFilter	query	Возвращать свойство `filter` с условиями корреляции.
selectOperations	query	Возвращать свойство `operations` с операциями корреляции.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `correlationid`, `name`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение корреляций

Получение всех настроенных корреляций вместе с условиями и операциями корреляции. Фильтр использует тип оценки "and/or", поэтому свойство `formula` пустое, а `eval_formula` генерируется автоматически.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "correlationid": "1",
            "name": "Correlation 1",
            "description": "",
            "status": "0",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
```

```
                        "type": "3",
                        "oldtag": "error",
                        "newtag": "ok",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "type": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Фильтр корреляции
- Операция корреляции

Источник

CCorrelation::get() в *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Описание

`object correlation.update(object/array correlations)`

Этот метод позволяет обновлять существующие корреляции.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства корреляции, которые необходимо обновить.

Для каждой корреляции должно быть определено свойство `correlationid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам корреляции, метод принимает следующие параметры.

Parameter	Type	Description
filter	object	Объект фильтра корреляции для замены текущего фильтра.
operations	array	Операции корреляции для замены существующих операций.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных корреляций в свойстве `correlationid`.

Примеры

Отключение корреляции

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
```

```
        "status": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Заменить условия, но сохранить метод оценки

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

См. также

- Фильтр корреляции
- Операция корреляции

Источник

Correlation::update() в *ui/include/classes/api/services/CCorrelation.php*.

Метрика APM

Этот класс предназначен для работы с метриками, собранными для мониторинга производительности приложений (APM).

Ссылки на объекты:

- Метрика APM
- Пример

Доступные методы:

- apmmetric.get - получение точек данных метрики

Объект метрики APM

Следующие объекты напрямую связаны с API apmmetric.

Note:
Каждое свойство, кроме type, соответствует столбцу в таблицах метрик (otel_metrics_gauge, otel_metrics_sum, otel_metrics_histogram и otel_metrics_exponential_histogram) источника данных телеметрии. Большинство свойств соответствуют столбцу с таким же именем (например, свойство metric_name соответствует столбцу MetricName). Если имена отличаются, столбцы источника данных указаны в таблицах ниже. Список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии.

Метрика APM

Объект метрики APM представляет точку данных метрики (значение или агрегированные значения измерения в определенный момент времени).

Объект метрики APM имеет следующие свойства:

Свойство	Тип	Описание
type	integer	Тип точки данных. Возможные значения: 0 - Gauge; 1 - Sum; 2 - Histogram; 3 - Exponential histogram. Тип определяется таблицей, в которой хранится точка данных: otel_metrics_gauge, otel_metrics_sum, otel_metrics_histogram или otel_metrics_exponential_histogram. Поведение свойства: - *только для чтения*
resource_attributes	object	Атрибуты ресурса точки данных в формате «ключ-значение» (например, host.name: eabf8c2a5ddc). Поведение свойства: - *только для чтения*
resource_schema_url	string	URL, если указан, схемы OpenTelemetry для атрибутов ресурса. Поведение свойства: - *только для чтения*
scope_name	string	Имя области инструментирования, создавшей точку данных. Поведение свойства: - *только для чтения*
scope_version	string	Версия области инструментирования, если указана. Поведение свойства: - *только для чтения*
scope_attributes	object	Атрибуты области инструментирования в формате «ключ-значение». Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
scope_schema_url	string	URL, если указан, схемы OpenTelemetry для области инструментирования и ее атрибутов. Поведение свойства: - *только для чтения*
service_name	string	Имя сервиса, создавшего точку данных (например, `zabbix-frontend`), полученное из атрибута ресурса OpenTelemetry `service.name`. Поведение свойства: - *только для чтения*
metric_name	string	Имя метрики, к которой относится точка данных (например, `http.server.request.duration`). Поведение свойства: - *только для чтения*
metric_description	string	Описание метрики, если указано. Поведение свойства: - *только для чтения*
metric_unit	string	Единица измерения значений метрики, если указана (например, `s` или `By`; см. UCUM). Поведение свойства: - *только для чтения*
attributes	object	Атрибуты точки данных в формате «ключ-значение» (например, `http.request.method: POST`). Поведение свойства: - *только для чтения*
start_time_unix	integer	Время Unix (в наносекундах), когда начался период измерения точки данных. Поведение свойства: - *только для чтения*
time_unix	integer	Время Unix (в наносекундах) точки данных. Для точек данных *Sum*, *Histogram* и *Exponential histogram* это конец периода измерения. Поведение свойства: - *только для чтения*
value	float	Значение точки данных. Применяется к точкам данных *Gauge* и *Sum*. По умолчанию: 0. Поведение свойства: - *только для чтения*
flags	integer	Битовая маска флагов точки данных. Если установлен младший значащий бит (значение 1), для точки данных значение не записывалось. Поведение свойства: - *только для чтения*
exemplars	array	Массив объектов exemplar, то есть отдельных измерений, сохраненных в качестве примеров, с трассировкой и спаном, во время которых они были записаны. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
aggregation_temporality	integer	Способ агрегирования значений точки данных во времени. Применяется к точкам данных *Sum*, *Histogram* и *Exponential histogram*. Возможные значения: 0 - *(по умолчанию)* Не указано; 1 - Delta (агрегирование за период с момента предыдущей точки данных); 2 - Cumulative (агрегирование за период с момента `start_time_unix`). Поведение свойства: - *только для чтения*
is_monotonic	boolean	Указывает, может ли сумма только увеличиваться или оставаться неизменной с течением времени. Применяется к точкам данных *Sum*. По умолчанию: `false`. Поведение свойства: - *только для чтения*
count	integer	Количество измерений в точке данных. Применяется к точкам данных *Histogram* и *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
sum	float	Сумма всех измерений в точке данных. Применяется к точкам данных *Histogram* и *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
bucket_counts	array	Массив целых чисел с количеством измерений в каждой корзине гистограммы. Корзины определяются параметром `explicit_bounds`. Применяется к точкам данных *Histogram*. По умолчанию: `[]`. Поведение свойства: - *только для чтения*
explicit_bounds	array	Массив чисел с плавающей точкой, задающих границы между корзинами гистограммы. Применяется к точкам данных *Histogram*. По умолчанию: `[]`. Поведение свойства: - *только для чтения*
min	float	Наименьшее из измерений в точке данных. Применяется к точкам данных *Histogram* и *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
max	float	Наибольшее из измерений в точке данных. Применяется к точкам данных *Histogram* и *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
scale	integer	Масштаб экспоненциальной гистограммы, определяющий границы корзин и разрешение. Более высокий масштаб обеспечивает более узкие корзины и более высокое разрешение; более низкий масштаб обеспечивает более широкие корзины и более низкое разрешение. Применяется к точкам данных *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
zero_count	integer	Количество измерений в нулевой корзине экспоненциальной гистограммы. Применяется к точкам данных *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
positive_bucket_counts	array	Массив целых чисел с количеством положительных измерений в каждой корзине экспоненциальной гистограммы. Границы корзин определяются параметром `scale`; индекс первой корзины задается в `positive_offset`. Применяется к точкам данных *Exponential histogram*. По умолчанию: `[]`. Поведение свойства: - *только для чтения*
positive_offset	integer	Индекс первой корзины в `positive_bucket_counts`. Применяется к точкам данных *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*
negative_bucket_counts	array	Массив целых чисел с количеством отрицательных измерений в каждой корзине экспоненциальной гистограммы. Границы корзин определяются параметром `scale`; индекс первой корзины задается в `negative_offset`. Применяется к точкам данных *Exponential histogram*. По умолчанию: `[]`. Поведение свойства: - *только для чтения*
negative_offset	integer	Индекс первой корзины в `negative_bucket_counts`. Применяется к точкам данных *Exponential histogram*. По умолчанию: 0. Поведение свойства: - *только для чтения*

Экземпляр

Объект экземпляра имеет следующие свойства:

Свойство	Тип	Описание
traceid	string	Идентификатор трассировки, если он есть, во время которой была записана измеренная величина. Столбец источника данных: `Exemplars.TraceId`.
spanid	string	Идентификатор интервала, если он есть, во время которого была записана измеренная величина. Столбец источника данных: `Exemplars.SpanId`.
value	float	Значение измеренной величины. Столбец источника данных: `Exemplars.Value`.
time_unix	integer	Время Unix (в наносекундах), когда была записана измеренная величина. Столбец источника данных: `Exemplars.TimeUnix`.
filtered_attributes	object	Атрибуты измеренной величины в формате «ключ-значение», которые не включены в `attributes` точки данных. Столбец источника данных: `Exemplars.FilteredAttributes`.

apmmetric.get

Описание

`integer/array apmmetric.get(object parameters)`

Метод позволяет получать точки данных метрик в соответствии с указанными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
time_from	timestamp	Возвращать только точки данных, время которых совпадает с указанным временем или является более поздним. Поведение параметра: - *обязательный*
time_till	timestamp	Возвращать только точки данных, время которых предшествует указанному времени. Поведение параметра: - *обязательный*
types	integer/array	Возвращать только точки данных указанных типов. Список поддерживаемых типов метрик см. на странице объекта метрики APM.
with_flags_on	integer	Возвращать только точки данных, у которых установлены все указанные флаги. Чтобы указать несколько флагов, используйте сумму их значений.
with_flags_off	integer	Возвращать только точки данных, у которых не установлен ни один из указанных флагов. Чтобы указать несколько флагов, используйте сумму их значений.

Параметр	Тип	Описание
resource_attributes	object/array	Возвращать только точки данных с указанными атрибутами ресурса. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все точки данных. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
resource_attributes_evaltype	integer	Метод вычисления условий в `resource_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
scope_attributes	object/array	Возвращать только точки данных с указанными атрибутами области действия. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все точки данных. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
scope_attributes_evaltype	integer	Метод вычисления условий в `scope_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
attributes	object/array	Возвращать только точки данных с указанными атрибутами. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все точки данных. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
attributes_evaltype	integer	Метод вычисления условий в `attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.

Параметр	Тип	Описание
filter	object	Возвращать только результаты, которые точно соответствуют указанному фильтру. Принимает объект, в котором ключами являются имена свойств, а значениями - либо отдельное значение, либо массив значений для сопоставления. Поддерживаемые свойства: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, в котором ключами являются имена свойств, а значениями - строки для поиска. Если дополнительные параметры не указаны, выполняется поиск `LIKE "%...%"`. Поддерживаемые свойства: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.
offset	integer	Количество записей, которые необходимо пропустить перед возвратом результата.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение точек данных гистограммы

Получите последнюю точку данных метрики-гистограммы `http.server.request.duration` службы `zabbix-frontend`.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
```

```
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Трассировка APM
- Сегмент APM
- Журнал APM

Источник

CApmMetric::get() в *ui/include/classes/api/services/CApmMetric.php*.

Модуль

Этот класс предназначен для работы с модулями веб-интерфейса.

Ссылки на объекты:

- Module

Доступные методы:

- module.create - установить новые модули
- module.delete - удалить модули
- module.get - получить модули
- module.update - обновить модули

Объект module

Следующие объекты напрямую связаны с API `module`.

Модуль

Объект module имеет следующие свойства.

Property	Type	Description
moduleid	ID	ID модуля, как он хранится в базе данных. Поведение свойства: - *только для чтения* - *обязательное* для операций обновления
id	string	Уникальный ID модуля, определенный разработчиком в файле `manifest.json` модуля. Возможные значения для встроенных модулей: см. описание свойства "type" в Dashboard widget. Поведение свойства: - *обязательное* для операций создания
relative_path	string	Путь к каталогу модуля относительно каталога веб-интерфейса Zabbix. Возможные значения: `widgets/*` - для встроенных модулей виджетов; `modules/*` - для сторонних модулей. Поведение свойства: - *обязательное* для операций создания
status	integer	Включен модуль или отключен. Возможные значения: 0 - *(по умолчанию)* Отключен; 1 - Включен.
config	object	Конфигурация модуля.

module.create

Описание

`object module.create(object/array modules)`

Этот метод позволяет устанавливать новые модули веб-интерфейса.

Note:
Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.
Если в файле конфигурации веб-интерфейса параметр `$ZBX_FEATURE_FLAGS['modules_config_enabled']` имеет значение `false`, пользователю не будет разрешено вызывать этот метод.

Attention:
Файлы модулей необходимо распаковать вручную в соответствующие подкаталоги согласно свойству `relative_path` модулей.

Параметры

`(object/array)` Модули для создания.

Метод принимает модули со стандартными свойствами модуля.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы установленных модулей в свойстве `moduleids`. Порядок возвращаемых идентификаторов соответствует порядку переданных модулей.

Примеры

Установка модуля

Установите модуль со статусом ”Enabled”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

См. также

- Модуль
- Модули веб-интерфейса

Источник

CModule::create() в *ui/include/classes/api/services/CModule.php*.

module.delete

Описание

`object module.delete(array moduleids)`

Этот метод позволяет удалить модули.

Note:
Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
 Если в файле конфигурации веб-интерфейса параметр `$ZBX_FEATURE_FLAGS['modules_config_enabled']` имеет значение `false`, пользователю не будет разрешено вызывать этот метод.

Attention:
Файлы модулей необходимо удалить вручную.

Параметры

`(array)` Идентификаторы модулей для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных модулей в свойстве `moduleids`.

Примеры

Удаление нескольких модулей

Удалите модули ”27” и ”28”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Источник

CModule::delete() в *ui/include/classes/api/services/CModule.php*.

module.get

Описание

`integer/array module.get(object parameters)`

Метод позволяет получить модули в соответствии с указанными параметрами.

Note:
Этот метод доступен только пользователям типа *Super admin*. Разрешение на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
Если в файле конфигурации веб-интерфейса параметр `$ZBX_FEATURE_FLAGS['modules_config_enabled']` установлен в значение `false`, пользователю не будет разрешено вызывать этот метод.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
moduleids	ID/array	Возвращать только модули с указанными ID.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `moduleid`, `relative_path`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(integer/array) Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр countOutput.

Примеры

Получение модуля по ID

Получить все данные о модулях "1", "2" и "25".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

См. также

- Модуль
- Виджет панели мониторинга
- Модули веб-интерфейса

Источник

CModule::get() в *ui/include/classes/api/services/CModule.php*.

module.update

Описание

`object module.update(object/array modules)`

Этот метод позволяет обновлять существующие модули.

> **Note:**
> Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
 Если в файле конфигурации веб-интерфейса параметр `$ZBX_FEATURE_FLAGS['modules_config_enabled']` имеет значение `false`, пользователю не будет разрешено вызывать этот метод.

Параметры

`(object/array)` Свойства модуля, которые необходимо обновить.

Свойство `moduleid` должно быть определено для каждого модуля, все остальные свойства необязательны. Будут обновлены только указанные свойства.

Метод принимает модули со стандартными свойствами модуля.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных модулей в свойстве `moduleids`.

Примеры

Отключение модуля

Отключить модуль "25".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

См. также

- Модуль
- Модули веб-интерфейса

Источник

CModule::update() в *ui/include/classes/api/services/CModule.php*.

Настройки

Этот класс предназначен для работы с общими административными настройками.

Ссылки на объекты:

- Settings

Доступные методы:

- settings.get - получить настройки
- settings.update - обновить настройки

Объект settings

Следующие объекты напрямую связаны с API `settings`.

Настройки

Объект настроек имеет следующие свойства.

Property	Type	Description
default_lang	string	Язык системы по умолчанию. По умолчанию: `en_US`.
default_timezone	string	Часовой пояс системы по умолчанию. По умолчанию: `system` - системный часовой пояс по умолчанию. Полный список поддерживаемых часовых поясов см. в документации PHP.
default_theme	string	Тема по умолчанию. Возможные значения: `blue-theme` - *(по умолчанию)* синяя; `blue-classic-theme` - синяя (классическая); `dark-theme` - темная; `dark-blue-theme` - темно-синяя; `dark-classic-theme` - темная (классическая); `hc-light` - светлая с высокой контрастностью; `hc-dark` - темная с высокой контрастностью.
search_limit	integer	Ограничение для результатов поиска и фильтрации. По умолчанию: 1000.
max_overview_table_size	integer	Максимальное количество столбцов и строк в виджетах панелей мониторинга «Обзор данных» и «Обзор триггеров». По умолчанию: 50.
max_in_table	integer	Максимальное количество элементов, отображаемых внутри ячейки таблицы. По умолчанию: 50.
server_check_interval	integer	Показывать предупреждение, если сервер Zabbix недоступен. Возможные значения: 0 - не показывать предупреждение; 10 - *(по умолчанию)* показывать предупреждение.
work_period	string	Рабочее время. По умолчанию: 1-5,09:00-18:00.
show_technical_errors	integer	Показывать технические ошибки (PHP/SQL) пользователям, не являющимся супер-администраторами, и пользователям, не входящим в группы пользователей с включенным режимом отладки. Возможные значения: 0 - *(по умолчанию)* не показывать технические ошибки; 1 - показывать технические ошибки.

Property	Type	Description
history_period	string	Максимальный период отображения данных истории в виджетах панелей мониторинга «Последние данные», «Веб» и «Обзор данных». Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 24h.
period_default	string	Период по умолчанию для фильтра времени. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d, 1M, 1y), включая поддержку месяцев и лет. По умолчанию: 1h.
max_period	string	Максимальный период для фильтра времени. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d, 1M, 1y), включая поддержку месяцев и лет. По умолчанию: 2y.
severity_color_0	string	Цвет уровня важности «Не классифицировано» в виде шестнадцатеричного кода цвета. По умолчанию: 97AAB3.
severity_color_1	string	Цвет уровня важности «Информация» в виде шестнадцатеричного кода цвета. По умолчанию: 7499FF.
severity_color_2	string	Цвет уровня важности «Предупреждение» в виде шестнадцатеричного кода цвета. По умолчанию: FFC859.
severity_color_3	string	Цвет уровня важности «Средняя» в виде шестнадцатеричного кода цвета. По умолчанию: FFA059.
severity_color_4	string	Цвет уровня важности «Высокая» в виде шестнадцатеричного кода цвета. По умолчанию: E97659.
severity_color_5	string	Цвет уровня важности «Катастрофа» в виде шестнадцатеричного кода цвета. По умолчанию: E45959.
severity_name_0	string	Название уровня важности «Не классифицировано». По умолчанию: Not classified.
severity_name_1	string	Название уровня важности «Информация». По умолчанию: Information.
severity_name_2	string	Название уровня важности «Предупреждение». По умолчанию: Warning.
severity_name_3	string	Название уровня важности «Средняя». По умолчанию: Average.
severity_name_4	string	Название уровня важности «Высокая». По умолчанию: High.
severity_name_5	string	Название уровня важности «Катастрофа». По умолчанию: Disaster.

Property	Type	Description
custom_color	integer	Использовать пользовательские цвета статусов событий. Возможные значения: 0 - *(по умолчанию)* не использовать пользовательские цвета статусов событий; 1 - использовать пользовательские цвета статусов событий.
ok_period	string	Период отображения триггеров со статусом OK. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h). По умолчанию: 5m.
blink_period	string	Период мигания при изменении состояния триггера. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h). По умолчанию: 2m.
problem_unack_color	string	Цвет неподтвержденных событий PROBLEM в виде шестнадцатеричного кода цвета. По умолчанию: CC0000.
problem_ack_color	string	Цвет подтвержденных событий PROBLEM в виде шестнадцатеричного кода цвета. По умолчанию: CC0000.
ok_unack_color	string	Цвет неподтвержденных событий RESOLVED в виде шестнадцатеричного кода цвета. По умолчанию: 009900.
ok_ack_color	string	Цвет подтвержденных событий RESOLVED в виде шестнадцатеричного кода цвета. По умолчанию: 009900.
problem_unack_style	integer	Мигание для неподтвержденных событий PROBLEM. Возможные значения: 0 - не показывать мигание; 1 - *(по умолчанию)* показывать мигание.
problem_ack_style	integer	Мигание для подтвержденных событий PROBLEM. Возможные значения: 0 - не показывать мигание; 1 - *(по умолчанию)* показывать мигание.
ok_unack_style	integer	Мигание для неподтвержденных событий RESOLVED. Возможные значения: 0 - не показывать мигание; 1 - *(по умолчанию)* показывать мигание.
ok_ack_style	integer	Мигание для подтвержденных событий RESOLVED. Возможные значения: 0 - не показывать мигание; 1 - *(по умолчанию)* показывать мигание.
url	string	URL веб-интерфейса.
discovery_groupid	ID	ID группы узлов сети, в которую будут автоматически помещаться обнаруженные узлы сети.
default_inventory_mode	integer	Режим инвентаризации узла сети по умолчанию. Возможные значения: -1 - *(по умолчанию)* отключен; 0 - ручной; 1 - автоматический.

Property	Type	Description
alert_usrgrpid	ID	ID группы пользователей, которой будет отправлено сообщение о недоступности базы данных. Если задано значение «0», сообщение отправлено не будет.
snmptrap_logging	integer	Записывать в журнал несопоставленные SNMP-трапы. Возможные значения: 0 - не записывать в журнал несопоставленные SNMP-трапы; 1 - *(по умолчанию)* записывать в журнал несопоставленные SNMP-трапы.
login_attempts	integer	Количество неудачных попыток входа, после которого форма входа будет заблокирована. По умолчанию: 5.
login_block	string	Интервал времени, в течение которого форма входа будет заблокирована, если количество неудачных попыток входа превысит значение, заданное в поле login_attempts. Принимает количество секунд (например, 1, 60) или значение с суффиксом времени (например, 30s, 1m). По умолчанию: 30s.
validate_uri_schemes	integer	Проверять схемы URI. Возможные значения: 0 - не проверять; 1 - *(по умолчанию)* проверять.
uri_valid_schemes	string	Допустимые схемы URI. По умолчанию: http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	HTTP-заголовок X-Frame-Options. По умолчанию: SAMEORIGIN.
iframe_sandboxing_enabled	integer	Использовать песочницу iframe. Возможные значения: 0 - не использовать; 1 - *(по умолчанию)* использовать.
iframe_sandboxing_exceptions	string	Исключения для песочницы iframe.
connect_timeout	string	Тайм-аут подключения к серверу Zabbix. Допустимый диапазон значений: 1-30s. По умолчанию: 3s. Поведение свойства: - *обязательное*
socket_timeout	string	Сетевой тайм-аут по умолчанию. Допустимый диапазон значений: 1-300s. По умолчанию: 3s. Поведение свойства: - *обязательное*
media_type_test_timeout	string	Сетевой тайм-аут для проверки типа носителя. Допустимый диапазон значений: 1-300s. По умолчанию: 65s. Поведение свойства: - *обязательное*

Property	Type	Description
item_test_timeout	string	Сетевой тайм-аут для проверки элементов данных. Допустимый диапазон значений: 1-600s. По умолчанию: 60s. Поведение свойства: - *обязательное*
script_timeout	string	Сетевой тайм-аут выполнения скрипта. Допустимый диапазон значений: 1-300s. По умолчанию: 60s. Поведение свойства: - *обязательное*
report_test_timeout	string	Сетевой тайм-аут проверки запланированного отчета. Допустимый диапазон значений: 1-300s. По умолчанию: 60s. Поведение свойства: - *обязательное*
device_link_timeout	string	Тайм-аут добавления мобильного устройства для использования с мобильным приложением Zabbix; применяется к обмену данными между API Zabbix и адаптером моста. Допустимый диапазон значений: 1-300s. По умолчанию: 60s. Поведение свойства: - *обязательное*
auditlog_enabled	integer	Включать ли аудит. Возможные значения: 0 - отключить; 1 - *(по умолчанию)* включить.
auditlog_mode	integer	Включать ли аудит действий низкоуровневого обнаружения, сетевого обнаружения и автоматической регистрации, выполняемых сервером (пользователем System). Возможные значения: 0 - отключить; 1 - *(по умолчанию)* включить.
ha_failover_delay	string	Задержка переключения в секундах. По умолчанию: 1m. Поведение свойства: - *только для чтения*

Property	Type	Description
geomaps_tile_provider	string	Поставщик тайлов географической карты. Возможные значения: `OpenStreetMap.Mapnik` - *(по умолчанию)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery. Поддерживает пустую строку для указания пользовательских значений `geomaps_tile_url`, `geomaps_max_zoom` и `geomaps_attribution`.
geomaps_tile_url	string	URL тайлов географической карты. Поведение свойства: - *поддерживается*, если для `geomaps_tile_provider` задана пустая строка
geomaps_max_zoom	integer	Максимальный уровень масштабирования географической карты. Возможные значения: 0-30. Поведение свойства: - *поддерживается*, если для `geomaps_tile_provider` задана пустая строка
geomaps_attribution	string	Текст атрибуции географической карты. Поведение свойства: - *поддерживается*, если для `geomaps_tile_provider` задана пустая строка
vault_provider	integer	Поставщик Vault. Возможные значения: 0 - *(по умолчанию)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Разрешать значения макросов секретов с помощью: Возможные значения: 0 - *(по умолчанию)* только сервера; 1 - сервера и прокси независимо.
timeout_zabbix_agent	string	Тратить на обработку не более `timeout_*` секунд. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, `30s`, `1m`). Также принимает пользовательские макросы. Допустимый диапазон значений: 1-600s. По умолчанию: 3s. По умолчанию для `timeout_browser`: 60s. Поведение свойства: - *обязательное*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

Property	Type	Description
timeout_telemetry_query		
apm_global_db	object	Конфигурация глобального источника данных.

Глобальный источник данных

Объект глобального источника данных имеет следующие свойства:

Свойство	Тип	Описание
status	integer	Состояние глобального источника данных. Возможные значения: 0 - *(по умолчанию)* не настроен; 1 - настроен.
url	string	URL-адрес, используемый для подключения к источнику данных. Поведение свойства: - *обязательное*, если пользователь изменяет `status` на 1 (настроен).
authentication_type	integer	Тип аутентификации. Возможные значения: 0 - *(по умолчанию)* используется имя пользователя и пароль; 1 - аутентификация не используется. Поведение свойства: - *обязательное*, если пользователь изменяет `status` на 1 (настроен).
username	string	Имя пользователя, используемое для входа в источник данных. Поведение свойства: - *поддерживается*, если для `status` установлено значение 1 (настроен), а для `authentication_type` - значение 0 (имя пользователя и пароль). - *обязательное*, если пользователь изменяет `authentication_type` на 0 (имя пользователя и пароль).
password	string	Пароль, используемый для входа в источник данных. Значение этого параметра может быть пустым. Поведение свойства: - *поддерживается*, если для `status` установлено значение 1 (настроен), а для `authentication_type` - значение 0 (имя пользователя и пароль).
db	string	Имя базы данных. Значение этого параметра может быть пустым. Поведение свойства: - *поддерживается*, если для `status` установлено значение 1 (настроен).
ssl_verify_peer	integer	Проверка сертификата сервера. Возможные значения: 0 - *(по умолчанию)* отключена; 1 - включена. Поведение свойства: - *поддерживается*, если для `status` установлено значение 1 (настроен), а в `url` используется схема `https://`.

Свойство	Тип	Описание
ssl_verify_host	integer	Проверка узла сети. Возможные значения: 0 - отключена; 1 - *(по умолчанию)* включена. Поведение свойства: - *поддерживается*, если для `status` установлено значение 1 (настроен), а в `url` используется схема `https://`.

settings.get

Описание

`object settings.get(object parameters)`

Метод позволяет получить объект настроек в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает только один параметр.

Parameter	Type	Description
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

`(object)` Возвращает объект настроек..

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
```

```
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
```

```
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```

Источник

CSettings::get() в *ui/include/classes/api/services/CSettings.php*.

settings.update

Описание

`object settings.update(object settings)`

Этот метод позволяет обновлять существующие общие настройки.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Свойства настроек, которые нужно обновить.

Возвращаемые значения

`(array)` Возвращает массив с именами обновленных параметров.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Источник

CSettings::update() в *ui/include/classes/api/services/CSettings.php*.

Обнаруженная служба

Этот класс предназначен для работы с обнаруженными службами.

Ссылки на объекты:

- Обнаруженная служба

Доступные методы:

- dservice.get - получить обнаруженные службы

Обнаруженный объект службы

Следующие объекты напрямую связаны с API `dservice`.

Обнаруженная служба

> **Note:**
> Обнаруженные службы создаются сервером Zabbix и не могут быть изменены через API.

Объект обнаруженной службы содержит информацию о службе, обнаруженной правилом сетевого обнаружения на узле сети. Он имеет следующие свойства.

Property	Type	Description
dserviceid	ID	ID обнаруженной службы.
dcheckid	ID	ID проверки обнаружения, использованной для обнаружения службы.
dhostid	ID	ID обнаруженного узла сети, на котором работает служба.
dns	string	DNS узла сети, на котором работает служба.
ip	string	IP-адрес узла сети, на котором работает служба.
lastdown	timestamp	Время, когда обнаруженная служба в последний раз была недоступна.
lastup	timestamp	Время, когда обнаруженная служба в последний раз была доступна.
port	integer	Номер порта службы.
status	integer	Состояние службы. Возможные значения: 0 - служба доступна; 1 - служба недоступна.
value	string	Значение, возвращаемое службой при выполнении проверки обнаружения Zabbix agent, SNMPv1, SNMPv2 или SNMPv3.

dservice.get

Описание

`integer/array dservice.get(object parameters)`

Метод позволяет получать обнаруженные службы в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dserviceids	ID/array	Возвращать только обнаруженные сервисы с указанными ID.
dhostids	ID/array	Возвращать только обнаруженные сервисы, которые принадлежат указанным обнаруженным узлам сети.
dcheckids	ID/array	Возвращать только обнаруженные сервисы, которые были обнаружены указанными проверками обнаружения.
druleids	ID/array	Возвращать только обнаруженные сервисы, которые были обнаружены указанными правилами обнаружения.
selectDRules	query	Возвращать свойство `drules` с массивом правил обнаружения, которые обнаружили сервис.
selectDHosts	query	Возвращать свойство `dhosts` с массивом обнаруженных узлов сети, к которым принадлежит сервис.
selectHosts	query	Возвращать свойство `hosts` с узлами сети с тем же IP-адресом и прокси, что и у сервиса. Поддерживает `count`.
limitSelects	integer	Ограничивает число записей, возвращаемых подвыборками. Применяется к следующим подвыборкам: `selectHosts` - результат будет отсортирован по `hostid`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получить службы, обнаруженные на узле сети

Получить все обнаруженные службы, выявленные на обнаруженном узле сети ”11”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "dserviceid": "12",
            "dhostid": "11",
            "value": "",
            "port": "80",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650607",
            "dcheckid": "5",
            "ip": "192.168.1.134",
            "dns": "john.local"
        },
        {
            "dserviceid": "13",
            "dhostid": "11",
            "value": "",
            "port": "21",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650610",
            "dcheckid": "6",
            "ip": "192.168.1.134",
            "dns": "john.local"
        }
    ],
    "id": 1
}
```


См. также

- Обнаруженный узел сети
- Проверка обнаружения
- Узел сети

Источник

CDService::get() в *ui/include/classes/api/services/CDService.php*.

Обнаруженный узел сети

Этот класс предназначен для работы с обнаруженными узлами сети.

Ссылки на объекты:

- Обнаруженный узел сети

Доступные методы:

- dhost.get - получить обнаруженные узлы сети

Объект обнаруженного узла сети

Следующие объекты напрямую связаны с API `dhost`.

Обнаруженный узел сети

> **Note:**
> Обнаруженные узлы сети создаются сервером Zabbix и не могут быть изменены через API.

Объект обнаруженного узла сети содержит информацию об узле сети, обнаруженном правилом сетевого обнаружения. Он имеет следующие свойства.

Property	Type	Description
dhostid	ID	ID обнаруженного узла сети.
druleid	ID	ID правила обнаружения, которое обнаружило узел сети.

Property	Type	Description
lastdown	timestamp	Время, когда обнаруженный узел сети в последний раз переходил в состояние down.
lastup	timestamp	Время, когда обнаруженный узел сети в последний раз переходил в состояние up.
status	integer	Находится ли обнаруженный узел сети в состоянии up или down. Узел сети считается up, если у него есть хотя бы один активный обнаруженный сервис. Возможные значения: 0 - узел сети up; 1 - узел сети down.

dhost.get

Описание

`integer/array dhost.get(object parameters)`

Метод позволяет получить обнаруженные узлы сети в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dhostids	ID/array	Возвращать только обнаруженные узлы сети с указанными ID.
druleids	ID/array	Возвращать только обнаруженные узлы сети, созданные указанными правилами обнаружения.
dserviceids	ID/array	Возвращать только обнаруженные узлы сети, на которых работают указанные службы.
selectDRules	query	Возвращать свойство `drules` с массивом правил обнаружения, которые обнаружили узел сети.
selectDServices	query	Возвращать свойство `dservices` с обнаруженными службами, работающими на узле сети. Поддерживает `count`.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectDServices` - результаты будут отсортированы по `dserviceid`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `dhostid`, `druleid`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parameter	Type	Description
startSearch	boolean	

Возвращаемые значения

(integer/array) Возвращает:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр countOutput.

Примеры

Получить обнаруженные узлы сети по правилу обнаружения

Получить все узлы сети и обнаруженные службы, которые они запускают и которые были обнаружены правилом обнаружения "4".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dservices": [
                {
                    "dserviceid": "1",
                    "dhostid": "1",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697227",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.1",
                    "dns": "station.company.lan"
                }
            ],
            "dhostid": "1",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697227",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "2",
                    "dhostid": "2",
                    "type": "4",
                    "key_": "",
                    "value": "",
```

```
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.4",
                "dns": "john.company.lan"
            }
        ],
        "dhostid": "2",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    },
    {
        "dservices": [
            {
                "dserviceid": "3",
                "dhostid": "3",
                "type": "4",
                "key_": "",
                "value": "",
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.26",
                "dns": "printer.company.lan"
            }
        ],
        "dhostid": "3",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    },
    {
        "dservices": [
            {
                "dserviceid": "4",
                "dhostid": "4",
                "type": "4",
                "key_": "",
                "value": "",
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.7",
                "dns": "mail.company.lan"
            }
        ],
        "dhostid": "4",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    }
],
```

```
    "id": 1
}
```

См. также

- Обнаруженная служба
- Правило обнаружения

Источник

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

Обслуживание

Этот класс предназначен для работы с обслуживанием.

Ссылки на объекты:

- Обслуживание

Доступные методы:

- housekeeping.get - получить данные обслуживания
- housekeeping.update - обновить данные обслуживания

Объект housekeeping

Следующие объекты напрямую связаны с API `housekeeping`.

Обслуживание

Объект настроек имеет следующие свойства.

Свойство	Тип	Описание
hk_events_mode	integer	Включить внутреннее обслуживание событий и оповещений. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_events_trigger	string	Период хранения данных триггеров. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 365d.
hk_events_service	string	Период хранения данных сервисов. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 1d.
hk_events_internal	string	Период хранения внутренних данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 1d.
hk_events_discovery	string	Период хранения данных обнаружения сети. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 1d.
hk_events_autoreg	string	Период хранения данных автоматической регистрации. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d).

Свойство	Тип	Описание
hk_services_mode	integer	Включить внутреннее обслуживание сервисов. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_services	string	Период хранения данных сервисов. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 365d.
hk_audit_mode	integer	Включить внутреннее обслуживание аудита. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_audit	string	Период хранения данных аудита. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 31d.
hk_sessions_mode	integer	Включить внутреннее обслуживание сеансов. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_sessions	string	Период хранения данных сеансов. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 31d.
hk_history_mode	integer	Включить внутреннее обслуживание истории. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_history_global	integer	Переопределить период хранения истории элементов данных. Возможные значения: 0 - Не переопределять; 1 - *(по умолчанию)* Переопределять.
hk_history	string	Период хранения данных истории. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 31d.
hk_trends_mode	integer	Включить внутреннее обслуживание трендов. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
hk_trends_global	integer	Переопределить период хранения трендов элементов данных. Возможные значения: 0 - Не переопределять; 1 - *(по умолчанию)* Переопределять.
hk_trends	string	Период хранения данных трендов. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 365d.

Свойство	Тип	Описание
db_extension	string	Флаг конфигурации расширения DB. Если этому флагу присвоено значение ”timescaledb”, сервер изменяет поведение при обслуживании и удалении элементов данных. Поведение свойства: - *только для чтения*
compression_availability	integer	Поддерживается ли сжатие данных базой данных (или ее расширением). Возможные значения: 0 - Недоступно; 1 - Доступно. Поведение свойства: - *только для чтения*
compression_status	integer	Включить сжатие истории и трендов в TimescaleDB. Возможные значения: 0 - *(по умолчанию)* Выключено; 1 - Включено.
compress_older	string	Сжимать записи истории и трендов, срок хранения которых превышает указанный период. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). По умолчанию: 7d.

housekeeping.get

Описание

`object housekeeping.get(object parameters)`

Метод позволяет получить объект housekeeping в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает только один параметр.

Parameter	Type	Description
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

`(object)` Возвращает объект очистки данных.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```

Источник

CHousekeeping ::get() в *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Описание

`object housekeeping.update(object housekeeping)`

Этот метод позволяет обновлять существующие настройки обслуживания данных.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Свойства очистки, которые необходимо обновить.

Возвращаемые значения

`(array)` Возвращает массив с именами обновленных параметров.

Примеры

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.update",
    "params": {
        "hk_events_mode": "1",
        "hk_events_trigger": "200d",
```

```
        "hk_events_internal": "2d",
        "hk_events_discovery": "2d"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Источник

CHousekeeping::update() в *ui/include/classes/api/services/CHousekeeping.php*.

Обслуживание

Этот класс предназначен для работы с обслуживанием.

Ссылки на объекты:

- Обслуживание
- Период времени
- Тег проблемы

Доступные методы:

- maintenance.create - создать новые обслуживания
- maintenance.delete - удалить обслуживания
- maintenance.get - получить обслуживания
- maintenance.update - обновить обслуживания

Объект обслуживания

Следующие объекты напрямую связаны с API `maintenance`.

Обслуживание

Объект обслуживания имеет следующие свойства.

Property	Type	Description
maintenanceid	ID	ID обслуживания. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя обслуживания. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
active_since	timestamp	Время, когда обслуживание становится активным (включительно). Указанное значение будет округлено вниз до минут. Поведение свойства: - *обязательно* для операций создания
active_till	timestamp	Время, когда обслуживание перестает быть активным (исключительно). Указанное значение будет округлено вниз до минут. Поведение свойства: - *обязательно* для операций создания
description	string	Описание обслуживания.
maintenance_type	integer	Тип обслуживания. Возможные значения: 0 - *(по умолчанию)* с сбором данных; 1 - без сбора данных.
tags_evaltype	integer	Метод оценки тегов проблемы. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.

Период времени

Объект периода времени используется для определения периодов, когда обслуживание должно вступить в силу. Он имеет следующие свойства.

Property	Type	Description
period	integer	Длительность периода обслуживания в секундах. Указанное значение будет округлено вниз до минут. Допустимый диапазон значений: 300 - 86399940. По умолчанию: 3600.
timeperiod_type	integer	Тип периода времени. Возможные значения: 0 - *(по умолчанию)* только один раз; 2 - ежедневно; 3 - еженедельно; 4 - ежемесячно.
start_date	timestamp	Дата, когда период обслуживания должен вступить в силу. Указанное значение будет округлено вниз до минут. По умолчанию: текущая дата. Property behavior: - *supported* if `timeperiod_type` is set to ”one time only”

Property	Type	Description
start_time	integer	Время суток, когда начинается обслуживание, в секундах. Указанное значение будет округлено вниз до минут. По умолчанию: 0. Property behavior: - *supported* if `timeperiod_type` is set to ”daily”, ”weekly”, or ”monthly”
every	integer	Для ежедневных и еженедельных периодов `every` определяет интервалы в днях или неделях, через которые обслуживание должно вступить в силу. Значение по умолчанию, если `timeperiod_type` установлено в ”daily” или ”weekly”: 1. Для ежемесячных периодов, когда задан `day`, свойство `every` определяет день месяца, когда обслуживание должно вступить в силу. Значение по умолчанию, если `timeperiod_type` установлено в ”monthly” и задан `day`: 1. Для ежемесячных периодов, когда задан `dayofweek`, свойство `every` определяет неделю месяца, когда обслуживание должно вступить в силу. Возможные значения, если `timeperiod_type` установлено в ”monthly” и задан `dayofweek`: 1 - *(по умолчанию)* первая неделя; 2 - вторая неделя; 3 - третья неделя; 4 - четвертая неделя; 5 - последняя неделя. Property behavior: - *supported* if `timeperiod_type` is set to ”daily”, ”weekly”, or ”monthly”
dayofweek	integer	Дни недели, когда обслуживание должно вступить в силу. Возможные битовые значения: 1 - понедельник; 2 - вторник; 4 - среда; 8 - четверг; 16 - пятница; 32 - суббота; 64 - воскресенье. Это поле битовой маски; допустима любая сумма возможных битовых значений (например, 21 для понедельника, среды и пятницы). Property behavior: - *required* if `timeperiod_type` is set to ”weekly” or if `timeperiod_type` is set to ”monthly” and `day` is not set
day	integer	День месяца, когда обслуживание должно вступить в силу. Property behavior: - *required* if `timeperiod_type` is set to ”monthly” and `dayofweek` is not set

Property	Type	Description
month	integer	Месяцы, когда обслуживание должно вступить в силу. Возможные битовые значения: 1 - январь; 2 - февраль; 4 - март; 8 - апрель; 16 - май; 32 - июнь; 64 - июль; 128 - август; 256 - сентябрь; 512 - октябрь; 1024 - ноябрь; 2048 - декабрь. Это поле битовой маски; допустима любая сумма возможных битовых значений (например, 585 для января, апреля, июля и октября). Property behavior: - *required* if `timeperiod_type` is set to ”monthly”

Имя события

Объект имени события используется для определения проблем, которые необходимо подавить после начала действия периода обслуживания. Имена событий можно указывать только в том случае, если для `maintenance_type` объекта обслуживания установлено значение ”with data collection”. Объект имеет следующие свойства.

Property	Type	Description
operator	integer	Оператор условия. Возможные значения: 2 - *(по умолчанию)* Содержит; 3 - Не содержит.
value	string	Значение имени события. Поведение свойства: - *обязательное*

Тег проблемы

Объект тега проблемы используется для определения проблем, которые необходимо подавлять после начала действия периода обслуживания. Теги можно указывать только в том случае, если для `maintenance_type` объекта обслуживания установлено значение ”with data collection”. Объект имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега проблемы. Поведение свойства: - *обязательное*
operator	integer	Оператор условия. Возможные значения: 0 - Равно; 1 - Не равно; 2 - *(по умолчанию)* Содержит; 3 - Не содержит.
value	string	Значение тега проблемы.

maintenance.create

Описание

`object maintenance.create(object/array maintenances)`

Этот метод позволяет создавать новые обслуживания.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Объекты обслуживания для создания.

Помимо стандартных свойств обслуживания, метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы узлов сети, которые необходимо перевести в режим обслуживания. Группы узлов сети должны содержать только свойство `groupid`. Поведение параметра: - *обязателен*, если `hosts` не задан
hosts	object/array	Узлы сети, которые необходимо перевести в режим обслуживания. Узлы сети должны содержать только свойство `hostid`. Поведение параметра: - *обязателен*, если `groups` не задан
triggers	object/array	Триггеры, которые необходимо перевести в режим обслуживания. Триггеры должны содержать только свойство `triggerid`. Поведение параметра: - *обязателен*, если `groups` и `hosts` не заданы - *поддерживается*, если maintenance_type=0 (сбор данных)
event_names	object/array	Имена событий для определения проблем, которые необходимо включить в обслуживание. Поведение параметра: - *поддерживается*, если maintenance_type=0 (сбор данных)
tags	object/array	Теги проблем для определения проблем, которые необходимо включить в обслуживание. Поведение параметра: - *поддерживается*, если `maintenance_type` объекта обслуживания установлен в значение «сбор данных»
timeperiods	object/array	Периоды времени обслуживания. Поведение параметра: - *обязателен*

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных обслуживаний, указанных в свойстве `maintenanceids`. Порядок возвращаемых ID совпадает с порядком переданных обслуживаний.

Примеры

Создание обслуживания

Создайте обслуживание со сбором данных для группы узлов сети с идентификатором ”20” и тегами проблем **service:mysqld** и **error**.
Оно должно быть активно с 17.03.2026 по 17.03.2027, начинаться каждое воскресенье в 18:00 и длиться один час.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Создание внепланового обслуживания

Создайте внеплановое обслуживание продолжительностью один час со сбором данных для группы узлов сети с идентификатором ”4”, триггера с идентификатором ”12345” и названием события **Printer: ink low**.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
```

```
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "4"
        ]
    },
    "id": 1
}
```

См. также

- Узел сети
- Группа узлов сети
- Триггер
- Имя события
- Тег проблемы
- Период времени

Источник

CMaintenance::create() в *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Описание

`object maintenance.delete(array maintenanceIds)`

Этот метод позволяет удалять периоды обслуживания.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых периодов обслуживания.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных периодов обслуживания, указанных в свойстве `maintenanceids`.

Примеры

Удаление нескольких периодов обслуживания

Удалите два периода обслуживания.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

Источник

CMaintenance::delete() в *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Описание

`integer/array maintenance.get(object parameters)`

Метод позволяет получать обслуживания в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
groupids	ID/array	Возвращать только периоды обслуживания, назначенные указанным группам узлов сети.
hostids	ID/array	Возвращать только периоды обслуживания, назначенные указанным узлам сети.
triggerids	ID/array	Возвращать только периоды обслуживания, назначенные указанным триггерам.
maintenanceids	ID/array	Возвращать только периоды обслуживания с указанными идентификаторами.
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, назначенными периоду обслуживания.

Параметр	Тип	Описание
selectHosts	query	Возвращать свойство `hosts` с узлами сети, назначенными периоду обслуживания.
selectTriggers	query	Возвращать свойство `triggers` с триггерами, назначенными периоду обслуживания.
selectEventNames	query	Возвращать свойство `event_names` с именами событий, назначенными периоду обслуживания.
selectTags	query	Возвращать свойство `tags` с тегами проблем периода обслуживания.
selectTimeperiods	query	Возвращать свойство `timeperiods` с временными периодами обслуживания.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение периодов обслуживания

Получите все настроенные периоды обслуживания, данные о назначенной области обслуживания (группа узлов сети, узел сети, триггер, имя события, теги) и заданные периоды времени.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
```

```
    "name": "Regular DB maintenance",
    "maintenance_type": "0",
    "description": "",
    "active_since": "1791936000",
    "active_till": "2107555200",
    "tags_evaltype": "0",
    "hostgroups": [
      {
        "groupid": "20",
        "name": "Databases",
        "flags": "0",
        "uuid": "748ad4d098d447d492bb935c907f652f"
      }
    ],
    "triggers": [],
    "event_names": [],
    "hosts": [],
    "tags": [],
    "timeperiods": [
      {
        "timeperiod_type": "4",
        "every": "1",
        "month": "4095",
        "dayofweek": "0",
        "day": "1",
        "start_time": "64800",
        "period": "3600",
        "start_date": "0"
      }
    ]
  },
  {
    "maintenanceid": "2",
    "name": "Ad-hoc: Printer: ink OUT",
    "maintenance_type": "0",
    "description": "",
    "active_since": "1792057740",
    "active_till": "1792061340",
    "tags_evaltype": "0",
    "hostgroups": [
      {
        "groupid": "4",
        "name": "Zabbix servers",
        "flags": "0",
        "uuid": "6f6799aa69e844b4b3918f779f2abf08"
      }
    ],
    "triggers": [],
    "event_names": [
      {
        "operator": "2",
        "value": "Printer: ink OUT"
      }
    ],
    "hosts": [],
    "tags": [],
    "timeperiods": [
      {
        "timeperiod_type": "0",
        "every": "1",
        "month": "0",
        "dayofweek": "0",
```

```
            "day": "0",
            "start_time": "0",
            "period": "3600",
            "start_date": "1792057740"
          }
        ]
      }
    ],
    "id": 1
}
```


См. также

- Узел сети
- Группа узлов сети
- Триггер
- Имя события
- Тег проблемы
- Период времени

Источник

CMaintenance::get() в *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Описание

`object maintenance.update(object/array maintenances)`

Этот метод позволяет обновлять существующие обслуживания.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Свойства обслуживания, которые необходимо обновить.

Для каждого обслуживания необходимо определить свойство `maintenanceid`; все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств обслуживания, метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы узлов сети, которыми необходимо заменить текущие группы. Для групп узлов сети должно быть определено только свойство `groupid`. Поведение параметра: - *обязателен*, если `hosts` не задан
hosts	object/array	Узлы сети, которыми необходимо заменить текущие узлы сети. Для узлов сети должно быть определено только свойство `hostid`. Поведение параметра: - *обязателен*, если `groups` не задан

Parameter	Type	Description
triggers	object/array	Триггеры, которыми необходимо заменить текущие триггеры. Для триггеров должно быть определено только свойство `triggerid`. Поведение параметра: - *обязателен*, если `groups` и `hosts` не заданы - *поддерживается*, если maintenance_type=0 (сбор данных)
event_names	object/array	Имена событий, которыми необходимо заменить текущие имена событий. Поведение параметра: - *поддерживается*, если maintenance_type=0 (сбор данных)
tags	object/array	Теги проблем, которыми необходимо заменить текущие теги. Поведение параметра: - *поддерживается*, если для `maintenance_type` объекта обслуживания установлено значение «сбор данных»
timeperiods	object/array	Периоды времени обслуживания, которыми необходимо заменить текущие периоды.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных обслуживаний, указанных в свойстве `maintenanceids`.

Примеры

Назначение других узлов сети

Замените узлы сети, которые в настоящее время назначены на обслуживание, на два других.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

См. также

- Период времени

Источник

CMaintenance::update() в *ui/include/classes/api/services/CMaintenance.php*.

Отчет

Этот класс предназначен для работы с запланированными отчетами.

Ссылки на объекты:

- Отчет
- Пользователи
- Группы пользователей

Доступные методы:

- report.create - создать новые запланированные отчеты
- report.delete - удалить запланированные отчеты
- report.get - получить запланированные отчеты
- report.update - обновить запланированные отчеты

Объект отчета

Следующие объекты напрямую связаны с API `report`.

Отчет

Объект отчета имеет следующие свойства:

Свойство	Тип	Описание
reportid	ID	Идентификатор отчета. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
userid	ID	Идентификатор пользователя, создавшего отчет. Поведение свойства: - *обязательно* для операций создания
name	string	Уникальное имя отчета. Поведение свойства: - *обязательно* для операций создания
dashboardid	ID	Идентификатор панели, на основе которой создается отчет. Поведение свойства: - *обязательно* для операций создания
period	integer	Период, за который будет подготовлен отчет. Возможные значения: 0 - *(по умолчанию)* предыдущий день; 1 - предыдущая неделя; 2 - предыдущий месяц; 3 - предыдущий год.
cycle	integer	Периодичность повторения расписания. Возможные значения: 0 - *(по умолчанию)* ежедневно; 1 - еженедельно; 2 - ежемесячно; 3 - ежегодно.
start_time	integer	Время суток в секундах, когда отчет будет подготовлен к отправке. По умолчанию: 0.

Свойство	Тип	Описание
weekdays	integer	Дни недели для отправки отчета. Возможные значения битовой маски: 1 - понедельник; 2 - вторник; 4 - среда; 8 - четверг; 16 - пятница; 32 - суббота; 64 - воскресенье. Это поле является битовой маской; допустима любая сумма возможных значений битовой маски (например, 21 для понедельника, среды и пятницы). По умолчанию: 0. Поведение свойства: - *обязательно*, если для `cycle` установлено значение ”weekly”.
active_since	string	Дата начала. Возможные значения: пустая строка - *(по умолчанию)* не указано (сохраняется как 0); конкретная дата в формате YYYY-MM-DD (сохраняется как временная метка начала дня (00:00:00)).
active_till	string	Дата окончания. Возможные значения: пустая строка - *(по умолчанию)* не указано (сохраняется как 0); конкретная дата в формате YYYY-MM-DD (сохраняется как временная метка конца дня (23:59:59)).
subject	string	Тема сообщения отчета.
message	string	Текст сообщения отчета.
status	integer	Указывает, включен или отключен отчет. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
description	text	Описание отчета.
state	integer	Состояние отчета. Возможные значения: 0 - *(по умолчанию)* отчет еще не обработан; 1 - отчет создан и успешно отправлен всем получателям; 2 - не удалось создать отчет; поле ”info” содержит информацию об ошибке; 3 - отчет создан, но не удалось отправить его некоторым или всем получателям; поле ”info” содержит информацию об ошибке. Поведение свойства: - *только для чтения*
lastsent	timestamp	Временная метка Unix последнего успешно отправленного отчета. Поведение свойства: - *только для чтения*
info	string	Описание ошибки или дополнительная информация. Поведение свойства: - *только для чтения*

Пользователи

Объект users имеет следующие свойства:

Property	Type	Description
userid	ID	ID пользователя, которому нужно отправить отчет. Поведение свойства: - *required*
access_userid	ID	ID пользователя, от имени которого будет сгенерирован отчет. 0 - *(default)* Генерировать отчет от имени получателя.
exclude	integer	Следует ли исключить пользователя из списка рассылки. Возможные значения: 0 - *(default)* Включить; 1 - Исключить.

Группы пользователей

Объект групп пользователей имеет следующие свойства:

Property	Type	Description
usrgrpid	ID	ID группы пользователей, которой будет отправлен отчет. Поведение свойства: - *required*
access_userid	ID	ID пользователя, от имени которого будет сгенерирован отчет. 0 - *(по умолчанию)* Сгенерировать отчет от имени получателя.

report.create

Описание

`object report.create(object/array reports)`

Этот метод позволяет создавать новые запланированные отчеты.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Планируемые отчеты для создания.

В дополнение к стандартным свойствам планируемого отчета метод принимает следующие параметры.

Parameter	Type	Description
users	object/array	Пользователи, которым нужно отправить отчет. Поведение параметра: - *required* if `user_groups` is not set
user_groups	object/array	Группы пользователей, которым нужно отправить отчет. Поведение параметра: - *required* if `users` is not set

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных запланированных отчетов в свойстве `reportids`. Порядок возвращаемых идентификаторов соответствует порядку переданных запланированных отчетов.

Примеры

Создание запланированного отчета

Создайте еженедельный отчет, который будет формироваться за предыдущую неделю каждый понедельник-пятницу в 12:00 с 2021-04-01 по 2021-08-31.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "report.create",
    "params": {
        "userid": "1",
        "name": "Weekly report",
        "dashboardid": "1",
        "period": "1",
        "cycle": "1",
        "start_time": "43200",
        "weekdays": "31",
        "active_since": "2021-04-01",
        "active_till": "2021-08-31",
        "subject": "Weekly report",
        "message": "Report accompanying text",
        "status": "1",
        "description": "Report description",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

См. также

- Пользователи
- Группы пользователей

Источник

CReport::create() в *ui/include/classes/api/services/CReport.php*.

report.delete

Описание

`object report.delete(array reportids)`

Этот метод позволяет удалять запланированные отчеты.

> **Note:**
> Этот метод доступен только пользователям с типом *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе User roles.

Параметры

`(array)` Идентификаторы запланированных отчетов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных запланированных отчетов в свойстве `reportids`.

Примеры

Удаление нескольких запланированных отчетов

Удалите два запланированных отчета.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Источник

CReport::delete() в *ui/include/classes/api/services/CReport.php*.

report.get

Описание

`integer/array report.get(object parameters)`

Метод позволяет получать запланированные отчеты в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object) Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
reportids	ID/array	Возвращать только запланированные отчеты с указанными ID отчетов.
expired	boolean	Если установлено значение true, возвращать только просроченные запланированные отчеты.
selectUsers	query	Возвращать свойство users с пользователями, которым настроена отправка отчета.
selectUserGroups	query	Возвращать свойство user_groups с группами пользователей, которым настроена отправка отчета.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: reportid, name, status.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр countOutput.

Примеры

Получение данных отчета

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": {
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "           ",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
```

```
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "           ",
            "message": "                  ",
            "status": "1",
            "description": "         ",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
            "name": "          ",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "          ",
            "message": "                  ",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


См. также

- Пользователи

- Группы пользователей

Источник

CReport::get() в *ui/include/classes/api/services/CReport.php*.

report.update

Описание

`object report.update(object/array reports)`

Этот метод позволяет обновлять существующие запланированные отчеты.

Note:
Этот метод доступен только для пользователей с типом *Admin* и *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства запланированного отчета, которые необходимо обновить.

Свойство `reportid` должно быть определено для каждого запланированного отчета, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам запланированного отчета метод принимает следующие параметры.

Parameter	Type	Description
users	object/array	Пользователи, которые заменят текущих пользователей, назначенных запланированному отчету. Поведение параметра: - *required* если `user_groups` не задан
user_groups	object/array	Группы пользователей, которые заменят текущие группы пользователей, назначенные запланированному отчету. Поведение параметра: - *required* если `users` не задан

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных запланированных отчетов в свойстве `reportids`.

Примеры

Отключение запланированного отчета

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
```

```
        ]
    },
    "id": 1
}
```

См. также

- Пользователи
- Группы пользователей

Источник

CReport::update() в *ui/include/classes/api/services/CReport.php*.

Панель мониторинга

Этот класс предназначен для работы с панелями мониторинга.

Ссылки на объекты:

- Панель мониторинга
- Страница панели мониторинга
    - Виджет панели мониторинга
        * Поле виджета панели мониторинга
- Группа пользователей панели мониторинга
- Пользователь панели мониторинга

Доступные методы:

- dashboard.create - создать новые панели мониторинга
- dashboard.delete - удалить панели мониторинга
- dashboard.get - получить панели мониторинга
- dashboard.update - обновить панели мониторинга

Объект панели

Следующие объекты напрямую связаны с API панелей `dashboard`.

Панель

Объект панели имеет следующие свойства.

Property	Type	Description
dashboardid	ID	ID панели. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя панели. Поведение свойства: - *обязательно* для операций создания
userid	ID	ID пользователя, который является владельцем панели.
private	integer	Тип общего доступа к панели. Возможные значения: 0 - общедоступная панель; 1 - *(по умолчанию)* частная панель.
display_period	integer	Интервал слайд-шоу по умолчанию (в секундах). Возможные значения: 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 30.

Property	Type	Description
auto_start	integer	Автоматический запуск слайд-шоу. Возможные значения: 0 - не запускать слайд-шоу автоматически; 1 - *(по умолчанию)* запускать слайд-шоу автоматически.

Страница панели

Объект страницы панели имеет следующие свойства.

Property	Type	Description
dashboard_pageid	ID	ID страницы панели. Поведение свойства: - *только для чтения*
name	string	Имя страницы панели. По умолчанию: пустая строка.
display_period	integer	Интервал слайд-шоу панели (в секундах). Возможные значения: 0, 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 0 (будет использоваться интервал слайд-шоу по умолчанию).
widgets	array	Массив объектов виджета панели.

Виджет панели

Объект виджета панели имеет следующие свойства.

Property	Type	Description
widgetid	ID	ID виджета панели. Поведение свойства: - *только для чтения*

Property	Type	Description
type	string	Тип виджета панели. Возможные значения: actionlog - Журнал действий; clock - Часы; discovery - Статус обнаружения; favgraphs - Избранные графики; favmaps - Избранные карты; gauge - Индикатор; geomap - Геокарта; graph - График (классический); graphprototype - Прототип графика; honeycomb - Honeycomb; hostavail - Доступность узла сети; hostcard - Карточка узла сети; hostnavigator - Навигатор узлов сети; itemcard - Карточка элемента данных; itemhistory - История элемента данных; itemnavigator - Навигатор элементов данных; item - Значение элемента данных; map - Карта; navtree - Дерево навигации по карте; piechart - Круговая диаграмма; problemhosts - Узлы сети с проблемами; problems - Проблемы; problemsbysv - Проблемы по серьезности; scatterplot - Точечная диаграмма; slareport - Отчет SLA; svggraph - График; systeminfo - Системная информация; tophosts - Топ узлов сети; topitems - Топ элементов данных; toptriggers - Топ триггеров; trigover - Обзор триггеров; url - URL; web - Мониторинг веб-сайтов. Поведение свойства: - *обязательно*
name	string	Пользовательское имя виджета.
x	integer	Горизонтальная позиция от левой стороны панели. Возможные значения: от 0 до 71.
y	integer	Вертикальная позиция от верхней стороны панели. Возможные значения: от 0 до 63.
width	integer	Ширина виджета. Возможные значения: от 1 до 72.
height	integer	Высота виджета. Возможные значения: от 1 до 64.
view_mode	integer	Режим отображения виджета. Возможные значения: 0 - *(по умолчанию)* стандартный вид виджета; 1 - со скрытым заголовком;
fields	array	Массив объектов поле виджета панели. Поведение свойства: - см. отдельные виджеты в Поля виджета панели

Поле виджета панели

Объект поля виджета панели имеет следующие свойства.

Property	Type	Description
type	integer	Тип поля виджета. Возможные значения: 0 - Integer; 1 - String; 2 - Группа узлов сети; 3 - Узел сети; 4 - Элемент данных; 5 - Прототип элемента данных; 6 - График; 7 - Прототип графика; 8 - Карта; 9 - Сервис; 10 - SLA; 11 - Пользователь; 12 - Действие; 13 - Тип медиа. Property behavior: - *required*
name	string	Имя поля виджета. Возможные значения: см. Поля виджета панели. Property behavior: - *required*
value	mixed	Значение поля виджета в зависимости от типа. Возможные значения: см. Поля виджета панели. Property behavior: - *required*

Группа пользователей панели мониторинга

Список прав доступа к панели мониторинга на основе групп пользователей. Имеет следующие свойства.

Property	Type	Description
usrgrpid	ID	ID группы пользователей. Property behavior: - *required*
permission	integer	Тип уровня доступа. Возможные значения: 2 - только чтение; 3 - чтение и запись. Property behavior: - *required*

Пользователь панели мониторинга

Список прав доступа к панели мониторинга на основе пользователей. Он имеет следующие свойства.

Property	Type	Description
userid	ID	ID пользователя. Property behavior: - *required*
permission	integer	Тип уровня прав доступа. Возможные значения: 2 - только чтение; 3 - чтение и запись. Property behavior: - *required*

dashboard.create

Описание

`object dashboard.create(object/array dashboards)`

Этот метод позволяет создавать новые панели мониторинга.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Панели мониторинга для создания.

В дополнение к стандартным свойствам панели мониторинга, метод принимает следующие параметры.

Parameter	Type	Description
pages	array	Страницы панели мониторинга, которые необходимо создать для панели мониторинга. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Поведение параметра: - *required*
users	array	Пользователи панели мониторинга, для которых будут созданы общие доступы на панели мониторинга.
userGroups	array	Группы пользователей панели мониторинга, для которых будут созданы общие доступы на панели мониторинга.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных панелей, указанных в свойстве `dashboardids`. Порядок возвращаемых ID совпадает с порядком переданных панелей.

Примеры

Создание панели мониторинга

Создайте панель мониторинга с именем "My dashboard" с одним виджетом Problems с тегами и с использованием двух типов общего доступа (группа пользователей и пользователь) на одной странице панели мониторинга.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
```

```
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "service"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "zabbix_server"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Страница панели мониторинга
- Виджет панели мониторинга

- Поле виджета панели мониторинга
- Пользователь панели мониторинга
- Группа пользователей панели мониторинга

Источник

CDashboard::create() в *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Описание

`object dashboard.delete(array dashboardids)`

Этот метод позволяет удалять панели мониторинга.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых панелей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных панелей, указанных в свойстве `dashboardids`.

Примеры

Удаление нескольких панелей

Удаление двух панелей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Источник

CDashboard::delete() в *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Описание

`integer/array dashboard.get(object parameters)`

Метод позволяет получать дашборды в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dashboardids	ID/array	Возвращать только дашборды с указанными ID.
selectPages	query	Возвращать свойство `pages` со страницами дашборда, правильно упорядоченными.
selectUsers	query	Возвращать свойство `users` с пользователями, которым предоставлен общий доступ к дашборду.
selectUserGroups	query	Возвращать свойство `userGroups` с группами пользователей, которым предоставлен общий доступ к дашборду.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `dashboardid`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(` `/` `)` Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение дашборда по ID

Получение всех данных о дашбордах "1" и "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "1",
            "name": "Dashboard",
            "userid": "1",
            "private": "0",
            "display_period": "30",
            "auto_start": "1",
            "users": [],
            "userGroups": [],
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "9",
                            "type": "systeminfo",
                            "name": "",
                            "x": "12",
                            "y": "8",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "8",
                            "type": "problemsbysv",
                            "name": "",
                            "x": "12",
                            "y": "4",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "7",
                            "type": "problemhosts",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "6",
                            "type": "discovery",
                            "name": "",
                            "x": "6",
                            "y": "9",
                            "width": "18",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
```

```
                    {
                        "widgetid": "5",
                        "type": "web",
                        "name": "",
                        "x": "0",
                        "y": "9",
                        "width": "18",
                        "height": "4",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "4",
                        "type": "problems",
                        "name": "",
                        "x": "0",
                        "y": "3",
                        "width": "12",
                        "height": "6",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "3",
                        "type": "favmaps",
                        "name": "",
                        "x": "8",
                        "y": "0",
                        "width": "12",
                        "height": "3",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "1",
                        "type": "favgraphs",
                        "name": "",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "3",
                        "view_mode": "0",
                        "fields": []
                    }
                ]
            },
            {
                "dashboard_pageid": "2",
                "name": "",
                "display_period": "0",
                "widgets": []
            },
            {
                "dashboard_pageid": "3",
                "name": "Custom page name",
                "display_period": "60",
                "widgets": []
            }
        ]
    },
    {
        "dashboardid": "2",
```

```
            "name": "My dashboard",
            "userid": "1",
            "private": "1",
            "display_period": "60",
            "auto_start": "1",
            "users": [
                {
                    "userid": "4",
                    "permission": "3"
                }
            ],
            "userGroups": [
                {
                    "usrgrpid": "7",
                    "permission": "2"
                }
            ],
            "pages": [
                {
                    "dashboard_pageid": "4",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "10",
                            "type": "problems",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "2",
                                    "name": "groupids",
                                    "value": "4"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


См. также

- Страница панели мониторинга
- Виджет панели мониторинга
- Поле виджета панели мониторинга
- Пользователь панели мониторинга
- Группа пользователей панели мониторинга

Источник

CDashboard::get() в *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

Описание

`object dashboard.update(object/array dashboards)`

Этот метод позволяет обновлять существующие дашборды.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства панели мониторинга, которые нужно обновить.

Для каждой панели мониторинга должно быть определено свойство `dashboardid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Дополнительно к стандартным свойствам панели мониторинга, метод принимает следующие параметры.

Parameter	Type	Description
pages	array	Страницы панели мониторинга для замены существующих страниц панели мониторинга. Страницы панели мониторинга обновляются по свойству `dashboard_pageid`. Новые страницы панели мониторинга будут созданы для объектов без свойства `dashboard_pageid`, а существующие страницы панели мониторинга будут удалены, если они не используются повторно. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Будут обновлены только указанные свойства страниц панели мониторинга.
users	array	Пользователи панели мониторинга для замены существующих элементов.
userGroups	array	Группы пользователей панели мониторинга для замены существующих элементов.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных панелей, указанных в свойстве `dashboardids`.

Примеры

Переименование панели мониторинга

Переименуйте панель мониторинга в ”SQL server status”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Обновление страниц панели мониторинга

Переименуйте первую страницу панели мониторинга, замените виджеты на второй странице панели мониторинга и добавьте новую страницу в качестве третьей. Удалите все остальные страницы панели мониторинга.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Изменение владельца панели

Доступно только администраторам и супер-администраторам.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Страница панели мониторинга
- Виджет панели мониторинга
- Поле виджета панели мониторинга
- Пользователь панели мониторинга
- Группа пользователей панели мониторинга

Источник

CDashboard::update() в *ui/include/classes/api/services/CDashboard.php*.

Поля виджета панели мониторинга

На этой странице содержатся навигационные ссылки для параметров виджета панели мониторинга и возможных значений свойств соответствующих объектов dashboard widget field.

Чтобы просмотреть параметры и значения свойств для каждого виджета, используйте боковую панель для перехода на соответствующую страницу виджета.

1. Журнал действий

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Журнал действий* в методах `dashboard.create` и `dashboard.update`.

> **Attention:**
> Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Журнал действий*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Журнал действий* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Получатели	11	userids.0	Идентификатор пользователя. Примечание: чтобы настроить нескольких пользователей, создайте объект поля виджета панели для каждого пользователя, увеличивая число в имени свойства.

Параметр	тип	имя	значение
Действия	12	actionids.0	Идентификатор действия. Примечание: чтобы настроить несколько действий, создайте объект поля виджета панели для каждого действия, увеличивая число в имени свойства.
Типы носителей	13	mediatypeids.0	Идентификатор типа носителя. Примечание: чтобы настроить несколько типов носителей, создайте объект поля виджета панели для каждого типа носителя, увеличивая число в имени свойства.
Состояние	0	statuses.0	0 - Выполняется; 1 - Отправлено/выполнено; 2 - Ошибка. Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая число в имени свойства.
Строка поиска	1	message	Любое строковое значение.
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` - использовать селектор периода времени панели в качестве источника данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*.
От	1	time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан - *обязателен*, если задан `time_period.to`
До	1	time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан - *обязателен*, если задан `time_period.from`
Сортировать записи по	0	sort_triggers	3 - Времени (по возрастанию); 4 - *(по умолчанию)* Времени (по убыванию); 5 - Типу (по возрастанию); 6 - Типу (по убыванию); 7 - Состоянию (по возрастанию); 8 - Состоянию (по убыванию); 11 - Получателю (по возрастанию); 12 - Получателю (по убыванию).
Показывать строки	0	show_lines	Допустимый диапазон значений: от 1 до 100. По умолчанию: 25.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Action log*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Журнал действий*

Настройте виджет *Журнал действий*, который отображает 10 записей с подробностями операций действий, отсортированных по времени (по возрастанию). Кроме того, отображайте только сведения о тех операциях действий, которые пытались отправить электронное письмо пользователю "1", но не были выполнены успешно.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "actionlog",
                        "name": "Action log",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "sort_triggers",
                                "value": 3
                            },
                            {
                                "type": 11,
                                "name": "userids.0",
                                "value": 1
                            },
                            {
                                "type": 13,
                                "name": "mediatypeids.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "statuses.0",
                                "value": 2
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
```

```
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

2. Часы

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Часы* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Часы*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Clock*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.
Тип времени	0	time_type	0 - *(по умолчанию)* Местное время; 1 - Время сервера; 2 - Время узла сети.
Тип часов	0	clock_type	0 - *(по умолчанию)* Аналоговые; 1 - Цифровые.

Следующие параметры поддерживаются, если для *Тип времени* установлено значение ”Время узла сети”.

Parameter	type	name	value
Элемент данных	4	itemid.0	ID элемента данных. Поведение параметра: - *обязательно*

Следующие параметры поддерживаются, если для *Тип часов* установлено значение ”Цифровые”.

Parameter	type	name	value
Показывать	0	show.0	1 - Дата; 2 - *(по умолчанию)* Время; 3 - Часовой пояс. Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.

Расширенная конфигурация

Следующие параметры расширенной конфигурации поддерживаются, если для *Clock type* установлено значение ”Digital”.

Parameter	type	name	value
Background color	1	bg_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Дата

Следующие параметры расширенной настройки поддерживаются, если для *Clock type* установлено значение ”Digital”, а для *Show* — ”Date”.

Parameter	type	name	value
Bold	0	date_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Color	1	date_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Время

Следующие параметры расширенной настройки поддерживаются, если для *Clock type* установлено значение ”Digital”, а для *Show* — значение ”Time”.

Parameter	type	name	value
Bold	0	time_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Color	1	time_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Seconds	0	time_sec	0 - Отключено; 1 - *(по умолчанию)* Включено.
Format	0	time_format	0 - *(по умолчанию)* 24-часовой формат; 1 - 12-часовой формат.

Часовой пояс

Следующие параметры расширенной конфигурации поддерживаются, если для *Clock type* установлено значение ”Digital”, а для *Show* — ”Time zone”.

Parameter	type	name	value
Bold	0	tzone_bold	0 - *(default)* Disabled; 1 - Enabled.
Color	1	tzone_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Time zone	1	tzone_timezone	Допустимая строка часового пояса (например, Europe/Riga, system, UTC и т. д.). Полный список поддерживаемых часовых поясов см. в документации PHP. По умолчанию: local. Поведение параметра: - *supported* если для *Time type* установлено значение "Local time" или "Server time"
Format	0	tzone_format	0 - *(default)* Short; 1 - Full. Поведение параметра: - *supported* если для *Time type* установлено значение "Local time" или "Server time"

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Clock*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Clock*

Настройте виджет *Clock*, который отображает локальные дату, время и часовой пояс в виде настраиваемых цифровых часов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "clock",
                        "name": "Clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "clock_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
```

```
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "date_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 0,
                                "name": "time_bold",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tzone_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "tzone_timezone",
                                "value": "Europe/Riga"
                            },
                            {
                                "type": 0,
                                "name": "tzone_format",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
```

```
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update

3 Статус обнаружения

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Состояние обнаружения* в методах dashboard.create и dashboard.update.

Параметры

Для виджета *Discovery status* поддерживаются следующие параметры.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Discovery status*. Дополнительные сведения о настройке панели мониторинга см. в dashboard.create.

Настройка виджета *Discovery status*

Настройте виджет *Discovery status* с интервалом обновления 15 минут.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "discovery",
                        "name": "Discovery status",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 900
```

```
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

4 Избранные графики

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Избранные графики* в методах `dashboard.create` и `dashboard.update`.

Параметры

Для виджета *Избранные графики* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Избранные графики*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Избранные графики*

Настройте виджет *Избранные графики* с интервалом обновления, установленным на 10 минут.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favgraphs",
                        "name": "Favorite graphs",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели

- dashboard.create
- dashboard.update

5 Избранные карты

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Избранные карты* в методах dashboard.create и dashboard.update.

Параметры

Следующие параметры поддерживаются для виджета *Избранные карты*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Favorite maps*. Дополнительную информацию о настройке панели мониторинга см. в dashboard.create.

Настройка виджета *Избранные карты*

Настройте виджет *Избранные карты* с интервалом обновления, установленным на 10 минут.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favmaps",
                        "name": "Favorite maps",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
```

```
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

6 Шкала

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Индикатор* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также повышает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Индикатор*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Gauge*.

Parameter		type	name
Refresh interval		0	rf_rate
Item		4	itemid.0
	Item (Widget)	1	itemid._reference
Min		1	min
Max		1	max
Value arc		1	value_arc_color
Arc background		1	empty_color
Background		1	bg_color
Show		0	show.0
Override host		1	override_hostid._reference

Расширенная настройка

Для виджета *Gauge* поддерживаются следующие параметры расширенной настройки.

Note:
Число в имени свойства *Thresholds* (например, thresholds.0.color) указывает позицию порога в списке, отсортированном по возрастанию. Однако если пороги настроены в другом порядке, после обновления конфигурации виджета в веб-интерфейсе Zabbix значения будут отсортированы по возрастанию (например, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Параметр	type	name	value
Угол	0	angle	Допустимые значения: 180 *(по умолчанию)* или 270.
Описание			
Описание	1	description	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {ITEM.*}, {INVENTORY.*}, пользовательские макросы. По умолчанию: {ITEM.NAME}.
Размер	0	desc_size	Допустимые значения: от 1 до 100. По умолчанию: 15.
Вертикальное положение	0	desc_v_pos	0 - Сверху; 1 - *(по умолчанию)* Снизу.
Полужирный шрифт	0	desc_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	desc_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Значение			
Десятичные разряды	0	decimal_places	Допустимые значения: от 1 до 10. По умолчанию: 2.
Размер	0	value_size	Допустимые значения: от 1 до 100. По умолчанию: 25.
Полужирный шрифт	0	value_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	value_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Единицы измерения			
Единицы измерения (флажок)	0	units_show	0 - Отключено; 1 - *(по умолчанию)* Включено.
Единицы измерения (значение)	1	units	Любое строковое значение. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение «Включено»
Размер	0	units_size	Допустимые значения: от 1 до 100. По умолчанию: 25. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение «Включено»

Параметр	type	name	value
Полужирный шрифт	0	units_bold	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение «Включено»
Положение	0	units_pos	0 - Перед значением; 1 - Над значением; 2 - *(по умолчанию)* После значения; 3 - Под значением. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение «Включено» Этот параметр игнорируется, если для него установлено одно из следующих временных единиц измерения: unixtime, uptime, s.
Цвет	1	units_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Дуга значения			
Размер дуги	0	value_arc_size	Допустимые значения: от 1 до 100. По умолчанию: 20.
Стрелка			
Цвет	1	needle_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто). Поведение параметра: - *поддерживается*, если для объекта поля виджета панели мониторинга *Показать* установлено значение «Дуга значения» или для параметра *Показать дугу* установлено значение «Включено»
Шкала			
Показывать единицы измерения	0	scale_show_units	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение «Включено» и для объекта поля виджета панели мониторинга *Показать* установлено значение «Дуга значения» либо для параметра *Показать дугу* установлено значение «Включено»
Размер	0	scale_size	Допустимые значения: от 1 до 100. По умолчанию: 15. Поведение параметра: - *поддерживается*, если для объекта поля виджета панели мониторинга *Показать* установлено значение «Дуга значения» или для параметра *Показать дугу* установлено значение «Включено»
Десятичные разряды	0	scale_decimal_places	Допустимые значения: от 1 до 10. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для объекта поля виджета панели мониторинга *Показать* установлено значение «Дуга значения» или для параметра *Показать дугу* установлено значение «Включено»
Пороги			
Цвет	1	thresholds.0.color	Шестнадцатеричный код цвета (например, FF0000).

Параметр	type	name	value
Порог	1	thresholds.0.threshold	Любое числовое значение. Поддерживаются суффиксы (например, «1d», «2w», «4K», «8G»).
Показывать подписи	0	th_show_labels	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если заданы *Пороги* и для объекта поля виджета панели мониторинга *Показать* установлено значение «Дуга значения» либо для параметра *Показать дугу* установлено значение «Включено»
Показывать дугу	0	th_show_arc	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если заданы *Пороги*
Размер дуги	0	th_arc_size	Допустимые значения: от 1 до 100. По умолчанию: 5. Поведение параметра: - *поддерживается*, если для параметра *Показывать дугу* установлено значение «Включено»

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Gauge*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Gauge*

Настройте виджет *Gauge*, который отображает значение элемента данных "44474" (Interface enp0s3: Bits sent). Кроме того, визуально доработайте виджет с помощью нескольких расширенных параметров, включая пороговые значения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "gauge",
                        "name": "Gauge",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 44474
                            },
                            {
                                "type": 1,
                                "name": "min",
                                "value": "100000"
```

```
                },
                {
                    "type": 1,
                    "name": "max",
                    "value": "1000000"
                },
                {
                    "type": 0,
                    "name": "show.0",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "show.1",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "show.2",
                    "value": 3
                },
                {
                    "type": 0,
                    "name": "show.4",
                    "value": 4
                },
                {
                    "type": 0,
                    "name": "show.5",
                    "value": 5
                },
                {
                    "type": 0,
                    "name": "angle",
                    "value": 270
                },
                {
                    "type": 0,
                    "name": "desc_size",
                    "value": 10
                },
                {
                    "type": 0,
                    "name": "desc_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "decimal_places",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "value_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "units_size",
                    "value": 15
                },
                {
```

```
                            "type": 0,
                            "name": "units_pos",
                            "value": 3
                        },
                        {
                            "type": 1,
                            "name": "needle_color",
                            "value": "3C3C3C"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.color",
                            "value": "FF465C"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.threshold",
                            "value": "700000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "FFD54F"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.threshold",
                            "value": "500000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.color",
                            "value": "0EC9AC"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.threshold",
                            "value": "100000"
                        },
                        {
                            "type": 0,
                            "name": "th_show_labels",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "th_show_arc",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "th_arc_size",
                            "value": 15
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
```

```
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

7 Геокарта

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Географическая карта* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Географическая карта*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Географическая карта* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.

Параметр	тип	имя	значение
Группы узлов сети	2	groupids.0	Идентификатор группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета на панели мониторинга для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо идентификатора группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Узлы сети	3	hostids.0	Идентификатор узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета на панели мониторинга для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо не настроен, либо настроен как минимум с одной группой узлов сети, в которую входят настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Узлы сети (виджет/панель мониторинга)	1	hostids._reference	Вместо идентификатора узла сети: `DASHBOARD.hostids` - задайте выбор узлов сети на панели мониторинга в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Теги			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/Или; 2 - Или. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов* Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.

Параметр	тип	имя	значение
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов* Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов* Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Начальный вид	1	default_view	Разделенные запятыми *широта*, *долгота*, *уровень масштабирования* (*необязательно*, допустимые значения от 0 до 30). Пример: `40.6892494,-74.0466891,10`.
Кластеризация			
Режим кластеризации	0	clustering_mode	0 - *(по умолчанию)* Автоматически; 1 - Уровень масштабирования.
Уровень масштабирования кластеризации	0	clustering_zoom_level	Допустимые значения от 0 до 30. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для параметра *Режим кластеризации* установлено значение «Уровень масштабирования»
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели мониторинга, которой принадлежит виджет. Поведение параметра: - *обязателен*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Geomap*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Geomap*

Настройте виджет *Geomap*, который отображает узлы сети из групп узлов сети ”2” и ”22” на основе следующей конфигурации тегов: тег с именем ”component” содержит значение ”node”, а тег с именем ”location” равен значению ”New York”. Кроме того, задайте начальный вид карты с координатами ”40.6892494” (широта), ”-74.0466891” (долгота) и уровнем масштабирования ”10”.

Запрос:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "geomap",
                        "name": "Geomap",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 22
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "default_view",
                                "value": "40.6892494,-74.0466891,10"
                            },
                            {
                                "type": 0,
                                "name": "evaltype",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "node"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "location"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 1
                            },
```

```
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "New York"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

8 График

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *График* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также повышает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *График*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Graph*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, к которой принадлежит виджет. Поведение параметра: - *required*

Набор данных

Для настройки *набора данных* поддерживаются следующие параметры.

Note:
Первое число в имени свойства (например, ds.0.hosts.0, ds.0.items.0) обозначает конкретный набор данных, а второе число, если оно указано, обозначает настроенный узел сети или элемент данных.

Параметр	тип	имя	значение
Тип набора данных	0	ds.0.dataset_type	0 - список элементов данных; 1 - *(по умолчанию)* шаблоны элементов данных.
Элементы данных	4	ds.0.itemids.0	Идентификатор элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне. Примечание: чтобы настроить несколько элементов данных, создайте отдельный объект поля виджета панели для каждого элемента данных, увеличивая число в имени свойства. Поведение параметра: - *обязателен*, если для параметра *Тип набора данных* установлено значение «Список элементов данных», а параметр *Элементы данных (виджет)* не задан
Элементы данных (виджет)	1	ds.0.itemids.0._reference	Вместо идентификатора элемента данных: `ABCDE._itemid` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для элементов данных. Примечание: чтобы настроить несколько виджетов, создайте отдельный объект поля виджета панели для каждого виджета, увеличивая число в имени свойства. Поведение параметра: - *обязателен*, если для параметра *Тип набора данных* установлено значение «Список элементов данных», а параметр *Элементы данных* не задан
Цвет	1	ds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *обязателен*, если для параметра *Тип набора данных* установлено значение «Список элементов данных»

Параметр	тип	имя	значение
Шаблоны узлов сети	1	ds.0.hosts.0	Имя или шаблон узла сети (например, ”Zabbix*”). Поведение параметра: - *обязателен*, если для параметра *Тип набора данных* установлено значение «Шаблоны элементов данных» Этот параметр не поддерживается при настройке виджета на панели шаблона.
Шаблоны элементов данных	1	ds.0.items.0	Имя или шаблон элемента данных (например, ”*: Number of processed *values per second”). При настройке виджета на панели шаблона следует указывать только шаблоны элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*, если для параметра *Тип набора данных* установлено значение «Шаблоны элементов данных»
Цвет	1	ds.0.color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для параметра *Тип набора данных* установлено значение «Шаблоны элементов данных», а параметр *Палитра цветов* не задан
Палитра цветов	0	ds.0.color_palette	Индекс палитры цветов. Допустимые значения находятся в диапазоне от 0 до 11. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для параметра *Тип набора данных* установлено значение «Шаблоны элементов данных», а параметр *Цвет* не задан
Отображение	0	ds.0.type	0 - *(по умолчанию)* линия; 1 - точки; 2 - ступенчатая линия; 3 - столбцы.
С накоплением	0	ds.0.stacked	0 - *(по умолчанию)* отключено; 1 - включено. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Линия», «Ступенчатая линия» или «Столбцы»
Ширина	0	ds.0.width	Допустимые значения находятся в диапазоне от 1 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Линия» или «Ступенчатая линия»
Размер точки	0	ds.0.pointsize	Допустимые значения находятся в диапазоне от 1 до 10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Точки»
Прозрачность	0	ds.0.transparency	Допустимые значения находятся в диапазоне от 1 до 10. По умолчанию: 5.

Параметр	тип	имя	значение
Заливка	0	ds.0.fill	Допустимые значения находятся в диапазоне от 1 до 10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Линия» или «Ступенчатая линия»
Отсутствующие данные	0	ds.0.missingdatafunc	0 - *(по умолчанию)* нет; 1 - соединять; 2 - считать равными 0; 3 - последнее известное значение. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Линия» или «Ступенчатая линия»
Переопределение узла сети	1	ds.0.override_hostid	`ABCDE` - задайте совместимый виджет (с параметром «Ссылка», установленным в значение "`ABCDE`") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задайте выбор узла сети панели в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Ось Y	0	ds.0.axisy	0 - *(по умолчанию)* левая; 1 - правая.
Инвертировать значения	0	ds.0.invert_values	0 - (по умолчанию) отключено; 1 - включено.
Сдвиг времени	1	ds.0.timeshift	Допустимая строка времени (например, `3600`, `1h` и т. д.). Можно использовать суффиксы времени. Также допускаются отрицательные значения. По умолчанию: "" (пусто).
Функция агрегации	0	ds.0.aggregate_function	0 - *(по умолчанию)* не используется; 1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма; 6 - первое значение; 7 - последнее значение.
Интервал агрегации	1	ds.0.aggregate_interval	Допустимая строка времени (например, `3600`, `1h` и т. д.). Можно использовать суффиксы времени. По умолчанию: `1h`.
Агрегировать	0	ds.0.aggregate_grouping	0 - *(по умолчанию)* каждый элемент данных; 1 - набор данных. Поведение параметра: - *поддерживается*, если для параметра *Функция агрегации* установлено значение «минимум», «максимум», «среднее», «количество», «сумма», «первое значение» или «последнее значение»
Аппроксимация	0	ds.0.approximation	1 - минимум; 2 - *(по умолчанию)* среднее; 4 - максимум; 7 - все.
Метка набора данных	1	ds.0.data_set_label	Любое строковое значение. По умолчанию: "" (пусто).
Теги элементов данных			

Параметр	тип	имя	значение
Тип оценки	0	ds.0.item_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	ds.0.item_tags.0.tag	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Оператор	0	ds.0.item_tags.0.operator	0 - содержит; 1 - равно; 2 - не содержит; 3 - не равно; 4 - существует; 5 - не существует. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Значение тега	1	ds.0.item_tags.0.value	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*

Параметры отображения

Для настройки *параметров отображения* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Выбор данных истории	0	source	0 - *(по умолчанию)* Автоматически; 1 - История; 2 - Тренды.
Простые триггеры	0	simple_triggers	0 - *(по умолчанию)* Отключено; 1 - Включено.
Рабочее время	0	working_time	0 - *(по умолчанию)* Отключено; 1 - Включено.
Имена узлов сети в подписях	0	show_hostnames	0 - *(по умолчанию)* Автоматически; 1 - Показывать; 2 - Скрывать. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Линия процентиля (слева)			
Состояние	0	percentile_left	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *оси Y* (в конфигурации *набора данных*) установлено значение «Слева»

Параметр	тип	имя	значение
Значение	0	percentile_left_value	Допустимый диапазон значений: от 1 до 100. Поведение параметра: - *поддерживается*, если для *оси Y* (в конфигурации *набора данных*) установлено значение «Слева»
Линия процентиля (справа)			
Состояние	0	percentile_right	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *оси Y* (в конфигурации *набора данных*) установлено значение «Справа»
Значение	0	percentile_right_value	Допустимый диапазон значений: от 1 до 100. Поведение параметра: - *поддерживается*, если для *оси Y* (в конфигурации *набора данных*) установлено значение «Справа»

Период времени

Следующие параметры поддерживаются для настройки *Периода времени*.

Parameter	type	name	value
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` - задать селектор периода времени на веб-интерфейсе панели как источник данных; `ABCDE._timeperiod` - задать совместимый виджет (с параметром *Reference*, установленным в значение "`ABCDE`") как источник данных. По умолчанию: `DASHBOARD._timeperiod` Кроме того, вы можете задать период времени только в параметрах *From* и *To*.
From	1	time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *supported* если *Период времени* не задан - *required* если задан `time_period.to`
To	1	time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *supported* если *Период времени* не задан - *required* если задан `time_period.from`

Оси

Следующие параметры поддерживаются для настройки *Оси*.

Parameter	type	name	value
Левая Y	0	lefty	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если *ось Y* (в настройке *Набор данных*) установлена в значение ”Left”
Правая Y	0	righty	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *ось Y* (в настройке *Набор данных*) установлена в значение ”Right”
Масштаб	0	lefty_scale righty_scale	0 - *(по умолчанию)* Линейный; 1 - Логарифмический.
Мин.	1	lefty_min righty_min	Любое числовое значение. По умолчанию: "" (пусто).
Макс.	1	lefty_max righty_max	Любое числовое значение. По умолчанию: "" (пусто).
Единицы (тип)	0	lefty_units righty_units	0 - *(по умолчанию)* Авто; 1 - Статические.
Единицы (значение)	1	lefty_static_units righty_static_units	Любое строковое значение. По умолчанию: "" (пусто).
Ось X	0	xaxis	0 - Отключено; 1 - *(по умолчанию)* Включено.

Легенда

Для настройки *Легенды* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Показывать легенду	0	legend	0 - Отключено; 1 - *(по умолчанию)* Включено.
Отображать мин./средн./макс.	0	legend_statistic	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Показывать функцию агрегации	0	legend_aggregation	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Строки	0	legend_lines_mode	0 - *(по умолчанию)* Фиксированное количество; 1 - Переменное количество. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»

Параметр	тип	имя	значение
Количество строк/ Максимальное количество строк	0	legend_lines	Допустимые значения: от 1 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Количество столбцов	0	legend_columns	Допустимые значения: от 1 до 4. По умолчанию: 4. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено», а для параметра *Отображать мин./средн./макс.* установлено значение «Отключено»

Проблемы

Следующие параметры поддерживаются для настройки *Проблем*.

Parameter	type	name	value
Показывать проблемы	0	show_problems	0 - *(по умолчанию)* Отключено; 1 - Включено.
Только выбранные элементы данных	0	graph_item_problems	0 - Отключено; 1 - *(по умолчанию)* Включено.
Узлы сети с проблемами	1	problemhosts.0	Имя узла сети. Примечание: Число в имени свойства указывает на настроенный узел сети. Чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая число в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Серьезность	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая число в имени свойства.
Проблема	1	problem_name	Имя события проблемы (без учета регистра, полное имя или его часть).
Теги проблемы			
Тип оценки	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.

Parameter	type	name	value
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает на порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*

Переопределения

Для настройки *переопределений* поддерживаются следующие параметры.

Note:
Первое число в имени свойства (например, or.0.hosts.0, or.0.items.0) обозначает определенный набор данных, а второе число, если оно указано, обозначает настроенный узел сети или элемент данных.

Параметр	тип	имя	значение
Шаблоны узлов сети	1	or.0.hosts.0	Имя или шаблон узла сети (например, `Zabbix*`). Этот параметр не поддерживается при настройке виджета на панели шаблона. Поведение параметра: - *обязателен* при настройке *переопределений*
Шаблоны элементов данных	1	or.0.items.0	Имя или шаблон элемента данных (например, `*: Number of processed *values per second`). При настройке виджета на панели шаблона следует указывать только шаблоны элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен* при настройке *переопределений*
Базовый цвет	1	or.0.color	Шестнадцатеричный код цвета (например, FF0000).
Ширина	0	or.0.width	Допустимые значения: от 1 до 10.
Отображение	0	or.0.type	0 - Линия; 1 - Точки; 2 - Ступенчатая линия; 3 - Столбцы.
Прозрачность	0	or.0.transparency	Допустимые значения: от 1 до 10.
Заливка	0	or.0.fill	Допустимые значения: от 1 до 10.
Размер точки	0	or.0.pointsize	Допустимые значения: от 1 до 10.

Параметр	тип	имя	значение
Отсутствующие данные	0	or.0.missingdatafunc	0 - Нет; 1 - Соединять; 2 - Считать равными 0; 3 - Последнее известное значение.
Ось Y	0	or.0.axisy	0 - Слева; 1 - Справа.
Инвертировать значения	0	or.0.invert_values	0 - (по умолчанию) Отключено; 1 - Включено.
Смещение времени	1	or.0.timeshift	Допустимая строка времени (например, 3600, 1h и т. д.). Можно использовать суффиксы времени. Допускаются отрицательные значения.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Graph*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Graph*

Настройте виджет *Graph* следующим образом:

- 2 набора данных для общего числа 9 элементов данных на 1 узле сети.
- Первый набор данных имеет тип "Item list" и состоит из 3 элементов, которые отображаются линиями разного цвета, но с одинаковыми шириной, прозрачностью и заливкой.
- Второй набор данных имеет тип "Item patterns", состоит из 6 элементов, имеет настроенную агрегацию и отображается линией с пользовательскими цветом, шириной, прозрачностью и заливкой.
- Второй набор данных также имеет пользовательскую метку набора данных.
- Данные на графике отображаются за период последних 3 часов.
- Проблемы на графике отображаются только для настроенных элементов данных.
- У графика есть две оси Y, при этом правая ось Y отображает значения только для второго набора данных.
- Легенда графика отображает настроенные элементы данных в 4 строках, а также минимальные, максимальные и средние значения наборов данных.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
```

```
                    {
                        "type": 1,
                        "name": "ds.0.color.0",
                        "value": "FF0000"
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.1",
                        "value": 23269
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.1",
                        "value": "BF00FF"
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.2",
                        "value": 23257
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.2",
                        "value": "0040FF"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.width",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "ds.0.transparency",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "ds.0.fill",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.1.hosts.0",
                        "value": "Zabbix server"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.items.0",
                        "value": "*: Number of processed *values per second"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.color",
                        "value": "000000"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.transparency",
                        "value": 0
                    },
                    {
                        "type": 0,
```

```
                "name": "ds.1.fill",
                "value": 0
            },
            {
                "type": 0,
                "name": "ds.1.axisy",
                "value": 1
            },
            {
                "type": 0,
                "name": "ds.1.aggregate_function",
                "value": 3
            },
            {
                "type": 1,
                "name": "ds.1.aggregate_interval",
                "value": "1m"
            },
            {
                "type": 0,
                "name": "ds.1.aggregate_grouping",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.1.data_set_label",
                "value": "Number of processed values per second"
            },
            {
                "type": 0,
                "name": "graph_time",
                "value": 1
            },
            {
                "type": 1,
                "name": "time_period.from",
                "value": "now-3h"
            },
            {
                "type": 1,
                "name": "time_period.to",
                "value": "now"
            },
            {
                "type": 0,
                "name": "legend_statistic",
                "value": 1
            },
            {
                "type": 0,
                "name": "legend_lines",
                "value": 4
            },
            {
                "type": 0,
                "name": "show_problems",
                "value": 1
            },
            {
                "type": 1,
                "name": "reference",
                "value": "YZABC"
```

```
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

9 График (классический)

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *График (классический)* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *График (классический)*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Graph (classic)*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Source	0	source_type	0 - *(по умолчанию)* График; 1 - Простой график.
Graph	6	graphid.0	ID Graph. Поведение параметра: - *required* если для *Source* задано значение "Graph"
Graph (Widget)	1	graphid._reference	Вместо ID Graph: `ABCDE._graphid` - укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для графиков. Поведение параметра: - *required* если для *Source* задано значение "Simple graph" и *Graph* не задан
Item	4	itemid.0	ID Item. Поведение параметра: - *required* если для *Source* задано значение "Simple graph" и *Item (Widget)* не задан
Item (Widget)	1	itemid._reference	Вместо ID Item: `ABCDE._itemid` - укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для элементов данных. Поведение параметра: - *required* если для *Source* задано значение "Simple graph" и *Item* не задан
Time period	1	time_period._reference	`DASHBOARD._timeperiod` - укажите селектор периода времени панели мониторинга в качестве источника данных; `ABCDE._timeperiod` - укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` Кроме того, можно задать период времени только в параметрах *From* и *To*.
From	1	time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *supported* если *Time period* не задан - *required* если задан `time_period.to`
To	1	time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *supported* если *Time period* не задан - *required* если задан `time_period.from`
Show legend	0	show_legend	0 - Отключено; 1 - *(по умолчанию)* Включено.

Parameter	type	name	value
Override host	1	override_hostid._reference	ABCDE._hostid - укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети; DASHBOARD._hostid - укажите селектор узла сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели мониторинга.
Reference	1	reference	Любое строковое значение, состоящее из 5 символов (например, ABCDE или JBPNL). Это значение должно быть уникальным в пределах панели мониторинга, к которой принадлежит виджет. Поведение параметра: - *required*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Graph (classic)*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Graph (classic)*

Настройте виджет *Graph (classic)*, который отображает простой график для элемента данных "42269".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "name": "Graph (classic)",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 1
                            },
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42269
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "RSTUV"
                            }
                        ]
                    }
```

```
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

10 Прототип графика

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Прототип графика* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Прототип графика*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Прототип графика* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.

Параметр	тип	имя	значение
Источник	0	source_type	2 - *(по умолчанию)* Прототип графика; 3 - Прототип простого графика.
Прототип графика	7	graphid.0	ID прототипа графика. Поведение параметра: - *обязателен*, если для параметра *Источник* установлено значение «Прототип графика»
Прототип элемента данных	5	itemid.0	ID прототипа элемента данных. Поведение параметра: - *обязателен*, если для параметра *Источник* установлено значение «Прототип простого графика»
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` - использовать селектор периода времени панели мониторинга в качестве источника данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*.
От	1	time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан - *обязателен*, если задан `time_period.to`
До	1	time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан - *обязателен*, если задан `time_period.from`
Показывать легенду	0	show_legend	0 - Отключено; 1 - *(по умолчанию)* Включено.
Переопределить узел сети	1	override_hostid._reference	`ABCDE._hostid` - использовать совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - использовать селектор узлов сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Столбцы	0	columns	Допустимые значения находятся в диапазоне от 1 до 24. По умолчанию: 2.
Строки	0	rows	Допустимые значения находятся в диапазоне от 1 до 16. По умолчанию: 1.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели мониторинга, которой принадлежит виджет. Поведение параметра: - *обязателен*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Graph prototype*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Graph prototype*

Настройте виджет *Graph prototype*, который отображает сетку из 3 графиков (3 столбца, 1 строка), созданных из прототипа элемента данных (ID: "42316") с помощью низкоуровневого обнаружения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graphprototype",
                        "name": "Graph prototype",
                        "x": 0,
                        "y": 0,
                        "width": 48,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 3
                            },
                            {
                                "type": 5,
                                "name": "itemid.0",
                                "value": 42316
                            },
                            {
                                "type": 0,
                                "name": "columns",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "OPQWX"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
```

```
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

11 Honeycomb

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Honeycomb* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Honeycomb*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Honeycomb*.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Параметр	type	name	value
Группы узлов сети (Виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Узлы сети	3	hostids.0	ID узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен либо вообще не быть настроен, либо быть настроен как минимум с одной группой узлов сети, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Узлы сети (Виджет/Панель)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Теги узлов сети			
Тип оценки	0	evaltype_host	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Оператор	0	host_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Параметр	type	name	value
Значение тега	1	host_tags.0.value	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Шаблоны элементов данных	1	items.0	Имя или шаблон элемента данных. Примечание: чтобы настроить несколько шаблонов элементов данных, создайте объект поля виджета панели для каждого шаблона элемента данных, увеличивая номер в имени свойства. Поведение параметра: - *обязательно*
Теги элементов данных			
Тип оценки	0	evaltype_item	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	item_tags.0.tag	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги элементов данных*
Оператор	0	item_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги элементов данных*
Значение тега	1	item_tags.0.value	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги элементов данных*
Показывать узлы сети в обслуживании	0	maintenance	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать	0	show.0	1 - Основная метка; 2 - Дополнительная метка. Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства. По умолчанию: 1, 2.

Параметр	type	name	value
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, ABCDE или JBPNL). Это значение должно быть уникальным в пределах панели, к которой принадлежит виджет. Поведение параметра: - *обязательно*

Расширенная настройка

Для виджета *Honeycomb* поддерживаются следующие расширенные параметры настройки.

Note:
Число в имени свойства *Пороговые значения* (например, thresholds.0.color) указывает позицию порогового значения в списке, отсортированном по возрастанию. Однако если пороговые значения настроены в другом порядке, после обновления конфигурации виджета в веб-интерфейсе Zabbix значения будут отсортированы по возрастанию (например, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Параметр	тип	имя	значение
Основная подпись			
Тип	0	primary_label_type	0 - *(по умолчанию)* Текст; 1 - Значение.
Текст	1	primary_label	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {ITEM.*}, {INVENTORY.*}, пользовательские макросы. По умолчанию: {HOST.NAME} Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Текст»
Десятичные разряды	0	primary_label_decimal_places	Допустимые значения: от 0 до 6. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение»
Размер (тип)	0	primary_label_size_type	0 - *(по умолчанию)* Автоматически; 1 - Пользовательский.
Размер	0	primary_label_size	Допустимые значения: от 1 до 100. По умолчанию: 20. Поведение параметра: - *поддерживается*, если для *Размер* (тип) установлено значение «Пользовательский»
Полужирный	0	primary_label_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	primary_label_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: зависит от параметра theme объекта настроек и объекта пользователя: 1F2C33 для «blue-theme», «blue-classic-theme» или «hc-light»; EEEEEE для «dark-theme», «dark-classic-theme» или «hc-dark»; E5F1FF для «dark-blue-theme».

Параметр	тип	имя	значение
Единицы измерения (флажок)	0	primary_label_units_show	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение»
Единицы измерения (значение)	1	primary_label_units	Любое строковое значение. "" (пусто) Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение», а для *Единицы измерения* (флажок) — «Включено»
Позиция	0	primary_label_units_pos	0 - Перед значением; 1 - *(по умолчанию)* После значения. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение», а для *Единицы измерения* (флажок) — «Включено» Этот параметр игнорируется, если ему присвоена одна из следующих единиц измерения времени: unixtime, uptime, s.
Вторичная подпись			
Тип	0	secondary_label_type	0 - Текст; 1 - *(по умолчанию)* Значение.
Текст	1	secondary_label	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {ITEM.*}, {INVENTORY.*}, пользовательские макросы. По умолчанию: {{ITEM.LASTVALUE}.fmtnum(2)} Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Текст»
Десятичные разряды	0	secondary_label_decimal_places	Допустимые значения: от 0 до 6. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение»
Размер (тип)	0	secondary_label_size_type	0 - *(по умолчанию)* Автоматически; 1 - Пользовательский.
Размер	0	secondary_label_size	Допустимые значения: от 1 до 100. По умолчанию: 30. Поведение параметра: - *поддерживается*, если для *Размер* (тип) установлено значение «Пользовательский»
Полужирный	0	secondary_label_bold	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	secondary_label_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: зависит от параметра theme объекта настроек и объекта пользователя: 1F2C33 для «blue-theme», «blue-classic-theme» или «hc-light»; EEEEEE для «dark-theme», «dark-classic-theme» или «hc-dark»; E5F1FF для «dark-blue-theme».
Единицы измерения (флажок)	0	secondary_label_units_show	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение»

Параметр	тип	имя	значение
Единицы измерения (значение)	1	secondary_label_units	Любое строковое значение. "" (пусто) Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение», а для *Единицы измерения* (флажок) — «Включено»
Позиция	0	secondary_label_position	0 - Перед значением; 1 - *(по умолчанию)* После значения. Поведение параметра: - *поддерживается*, если для *Тип* установлено значение «Значение», а для *Единицы измерения* (флажок) — «Включено» Этот параметр игнорируется, если ему присвоена одна из следующих единиц измерения времени: unixtime, uptime, s.
Цвет фона			
Цвет фона	1	bg_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: зависит от параметра theme объекта настроек и объекта пользователя: D9E7ED для «blue-theme» и «blue-classic-theme»; 3D5059 для «dark-theme» и «dark-classic-theme»; 004FA8 для «dark-blue-theme»; AAD7E9 для «hc-light»; 335463 для «hc-dark».
Пороговые значения			
Интерполяция цвета	0	interpolation	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	thresholds.0.color	Шестнадцатеричный код цвета (например, FF0000).
Пороговое значение	1	thresholds.0.threshold	Любое числовое значение. Поддерживаются суффиксы (например, «1d», «2w», «4K», «8G»).

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Honeycomb*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Honeycomb*

Настройте виджет *Honeycomb*, который отображает загрузку процессов сервера Zabbix. Кроме того, измените основную метку ячеек honeycomb и визуально настройте виджет с помощью пороговых значений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "honeycomb",
                        "name": "Honeycomb",
                        "x": "0",
                        "y": "0",
                        "width": "24",
                        "height": "5",
```

```
                    "view_mode": "0",
                    "fields": [
                        {
                            "type": 2,
                            "name": "groupids.0",
                            "value": 4
                        },
                        {
                            "type": 3,
                            "name": "hostids.0",
                            "value": 10084
                        },
                        {
                            "type": 1,
                            "name": "items.0",
                            "value": "Zabbix server: Utilization*"
                        },
                        {
                            "type": 1,
                            "name": "primary_label",
                            "value": "{ITEM.NAME}"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.color",
                            "value": "0EC9AC"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.threshold",
                            "value": "0"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "FFD54F"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.threshold",
                            "value": "70"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.color",
                            "value": "FF465C"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.threshold",
                            "value": "90"
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "KSTMQ"
                        }
                    ]
                }
            ]
        }
    ],
```

```
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

12 Доступность узла сети

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Доступность узлов сети* в методах `dashboard.create` и `dashboard.update`.

Параметры

Следующие параметры поддерживаются для виджета *Доступность узлов сети*.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Не обновлять; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Параметр	type	name	value
Группы узлов сети (виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Тип интерфейса	0	interface_type.0	0 - Нет; 1 - Zabbix agent (пассивные проверки); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix agent (активные проверки). По умолчанию: 1, 2, 3, 4, 5 (все включены). Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный.
Показывать узлы сети в обслуживании	0	maintenance	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать только итоги	0	only_totals	0 - *(по умолчанию)* Отключено; 1 - Включено.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Доступность узла сети*. Дополнительные сведения о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Доступность узла сети*

Настройте виджет *Доступность узла сети*, который отображает информацию о доступности (в вертикальном расположении) для узлов сети в группе узлов сети "4" с настроенными интерфейсами "Zabbix agent" и "SNMP".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostavail",
                        "name": "Host availability",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
```

```
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

13. Карточка узла сети

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Карточка узла сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карточка узла сети*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Карточка узла сети* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Не обновлять; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Узел сети	3	hostid.0	ID узла сети. Поведение параметра: - *обязательно*, если *Узел сети (Виджет/Панель)* не задан Этот параметр не поддерживается при настройке виджета на панели шаблона.
Узел сети (Виджет/Панель)	1	hostid._reference	Вместо ID узла сети: `DASHBOARD.hostid` — использовать селектор узла сети панели как источник данных для узла сети; `ABCDE._hostid` — использовать совместимый виджет (с его параметром *Ссылка*, установленным в "`ABCDE`") как источник данных для узла сети. Поведение параметра: - *обязательно*, если *Узел сети* не задан Этот параметр не поддерживается при настройке виджета на панели шаблона.
Показывать подавленные проблемы	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать	0	sections.0	0 - Группы узлов сети; 1 - Описание; 2 - Мониторинг; 3 - Доступность; 4 - Кем мониторится; 5 - Шаблоны; 6 - Инвентарные данные; 7 - Теги. Примечание: Число в имени свойства указывает на порядок раздела в списке разделов. Чтобы настроить несколько разделов, создайте объект поля виджета панели для каждого раздела с увеличивающимся номером в имени свойства.

Следующие параметры поддерживаются, если *Показывать* установлено в "Инвентарные данные".

Параметр	type	name	value
Поля инвентарных данных	0	inventory.0	ID инвентарных данных. Примечание: Чтобы настроить несколько полей инвентарных данных, создайте объект поля виджета панели для каждого поля инвентарных данных с увеличивающимся номером в имени свойства.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Карточка узла сети*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Карточка узла сети*

Настройте виджет *Карточка узла сети*, который отображает следующие разделы: «Мониторинг», «Доступность», «Контролируется», «Инвентарные данные» и «Метки».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostid.0",
                                "value": 10084
                            },
                            {
                                "type": 0,
                                "name": "show_suppressed",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 4
```

```
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 6
                            },
                            {
                                "type": 0,
                                "name": "sections.4",
                                "value": 7
                            },
                            {
                                "type": 0,
                                "name": "inventory.0",
                                "value": 25
                            },
                            {
                                "type": 0,
                                "name": "inventory.1",
                                "value": 26
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update

14. Навигатор по узлам сети

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Навигатор узлов сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Навигатор узлов сети*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Навигатор узлов сети* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	Идентификатор группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо идентификатора группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Шаблоны узлов сети	1	hosts.0	Имя или шаблон узла сети. Примечание: Чтобы настроить несколько шаблонов узлов сети, создайте объект поля виджета панели для каждого шаблона узла сети, увеличивая номер в имени свойства. Для нескольких шаблонов узлов сети параметр *Группы узлов сети* должен быть либо не настроен, либо настроен как минимум с одной группой узлов сети, к которой принадлежат узлы сети, соответствующие настроенным шаблонам узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Состояние узла сети	0	status	-1 - *(по умолчанию)* Любое; 0 - Включен; 1 - Отключен. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги узлов сети			
Тип вычисления	0	host_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	тип	имя	значение
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Оператор	0	host_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Значение тега	1	host_tags.0.value	Любое строковое значение. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Важность	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Показывать узлы сети в режиме обслуживания	0	maintenance	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать проблемы *Группировать по*	0	show_problems	0 - Все; 1 - *(по умолчанию)* Не подавленные; 2 - Нет.

Параметр	тип	имя	значение
Атрибут	0	group_by.0.attribute	0 - Группа узлов сети; 1 - Значение тега; 2 - Важность. Примечание: Номер в имени свойства указывает порядок атрибута в списке атрибутов группировки. Поведение параметра: - *обязателен* при настройке *Группировать по*
Значение	1	group_by.0.tag_name	Любое строковое значение. Примечание: Номер в имени свойства указывает набор атрибутов группировки, заданный в параметре *Атрибут*. Поведение параметра: - *обязателен* при настройке *Группировать по*, если для параметра *Атрибут* задано значение ”Значение тега”
Ограничение узлов сети	0	show_lines	Допустимые значения находятся в диапазоне от 1 до 9999. По умолчанию: 100. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, которой принадлежит виджет. Поведение параметра: - *обязателен*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Host navigator*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Host navigator*

Настройте виджет *Host navigator*, который отображает узлы сети, сгруппированные сначала по их группе узлов сети, а затем по значению тега ”city”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostnavigator",
                        "name": "Host navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
```

```
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "city"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "SWKLB"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели

- dashboard.create
- dashboard.update

15 Карточка элемента данных

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Карточка элемента данных* в методах dashboard.create и dashboard.update.

Attention:
Свойства fields виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также повышает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карточка элемента данных*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Карточка элемента данных* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Элемент данных	4	itemid.0	ID элемента данных. Поведение параметра: - *обязателен*, если *Элемент данных (Виджет)* не задан
Элемент данных (Виджет)	1	itemid._reference	Вместо ID элемента данных: ABCDE._itemid — задать совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для элемента данных. Поведение параметра: - *обязателен*, если *Элемент данных* не задан
Показывать	0	sections.0	0 - Описание; 1 - Текст ошибки; 2 - Метрики; 3 - Последние данные; 4 - Тип информации; 5 - Триггеры; 6 - Интерфейс узла сети; 7 - Тип; 8 - Инвентарные данные узла сети; 9 - Теги. Примечание: Число в имени свойства указывает порядок раздела в списке разделов. Чтобы настроить несколько разделов, создайте объект поля виджета панели с увеличивающимся числом в имени свойства для каждого раздела.

Параметр	type	name	value
Переопределить узел сети	1	override_hostid._reference	ABCDE._hostid — задать совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для узлов сети; DASHBOARD._hostid — задать селектор узла сети панели в качестве источника данных для узлов сети. По умолчанию: "" (пусто) Этот параметр не поддерживается при настройке виджета на панели шаблона.

Спарклайн

Следующие параметры поддерживаются, если для *Отображение* установлено значение «Последние данные».

Параметр	тип	имя	значение
Ширина	0	sparkline.width	Допустимые значения: от 0 до 10. По умолчанию: 1.
Заливка	0	sparkline.fill	Допустимые значения: от 0 до 10. По умолчанию: 3.
Цвет	1	sparkline.color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: 42A5F5.
Период времени	1	sparkline.time_period.reference	DASHBOARD._timeperiod - использовать селектор периода времени панели мониторинга в качестве источника данных; ABCDE._timeperiod - использовать совместимый виджет (с параметром reference, равным ABCDE) в качестве источника данных. По умолчанию: "" (пусто) Также можно задать период времени только в параметрах *От* и *До*.
От	1	sparkline.time_period.from	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или относительном формате (now, now/d, now/w-1w и т. д.). По умолчанию: now-1h. Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан; - *обязателен*, если задан sparkline.time_period.to
До	1	sparkline.time_period.to	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или относительном формате (now, now/d, now/w-1w и т. д.). По умолчанию: now. Поведение параметра: - *поддерживается*, если параметр *Период времени* не задан; - *обязателен*, если задан time_period.to
Данные истории	0	sparkline.history	0 - *(по умолчанию)* автоматически; 1 - история; 2 - тренды.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Item card*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *Карточка элемента данных*

Настройте виджет *Карточка элемента данных*, отображающий следующие разделы: «Описание», «Последние данные», «Триггеры» и «Теги».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
```

```
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

См. также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

16 История элемента данных

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *История элементов данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *История элементов данных*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *История элементов данных* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Расположение *Столбцы* (см. ниже)	0	layout	0 - *(по умолчанию)* Горизонтальное; 1 - Вертикальное.
Показывать строки	0	show_lines	Допустимый диапазон значений: от 1 до 100. По умолчанию: 25.

Параметр	тип	имя	значение
Переопределить узел сети	1	override_hostid._reference	`ABCDE._hostid` - задать совместимый виджет (с параметром *Ссылка*, установленным в "`ABCDE`") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задать селектор узлов сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Расширенная настройка (см. ниже)			
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели мониторинга, которой принадлежит виджет. Поведение параметра: - *обязательный*

Столбцы

Столбцы имеют общие параметры и дополнительные параметры в зависимости от настройки параметра *Item*.

Note:
Для всех параметров, связанных со столбцами, число в имени свойства (например, columns.0.name) указывает на столбец, для которого настроен параметр.

Следующие параметры поддерживаются для всех столбцов.

Parameter	type	name	value
Name	1	columns.0.name	Любое строковое значение. Поведение параметра: - *required*
Item	4	columns.0.itemid	ID элемента данных. При настройке виджета на шаблонной панели следует указывать только элементы данных, настроенные на шаблоне. Поведение параметра: - *required*
Base color	1	columns.0.base_color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: "" (пусто).

Поддерживаются следующие параметры столбца, если настроенный *Элемент данных* является элементом числового типа.

Параметр	тип	имя	значение
Отображение	0	columns.0.display	1 - *(по умолчанию)* Как есть; 2 - Полоса; 3 - Индикаторы.
Мин.	1	columns.0.min	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение "Полоса" или "Индикаторы"

Параметр	тип	имя	значение
Макс.	1	columns.0.max	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Полоса” или ”Индикаторы”
Пороги			
Цвет	1	columns.0.thresholds.color	Шестнадцатеричный код цвета (например, FF0000).
Порог	1	columns.0.thresholds.threshold	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”).
Данные истории	0	columns.0.history	0 - *(по умолчанию)* Авто; 1 - История; 2 - Тренды.

Следующие параметры столбцов поддерживаются, если настроенный *элемент данных* имеет тип «Символ», «Текст» или «Журнал».

Параметр	тип	имя	значение
Выделения			
Выделение	1	columns.0.highlights.color	Шестнадцатеричный код цвета (например, FF0000).
Пороговое значение	1	columns.0.highlights.pattern	Любое регулярное выражение.
Отображение	0	columns.0.display	1 - *(по умолчанию)* Как есть; 4 - HTML; 5 - Одна строка.
Одна строка	0	columns.0.max_length	Допустимые значения: от 1 до 500. По умолчанию: 100. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Одна строка»
Использовать моноширинный шрифт	0	columns.0.monospace_font	0 - *(по умолчанию)* Использовать шрифт по умолчанию; 1 - Использовать моноширинный шрифт.
Отображать местное время	0	columns.0.local_time	0 - *(по умолчанию)* Отображать временную метку; 1 - Отображать местное время. Поведение параметра: - *поддерживается*, если для *элемента данных* установлен тип «Журнал», а для параметра *Показывать временную метку* установлено значение «Включено»

Поддерживаются следующие параметры столбца, если настроенный *элемент данных* является элементом данных бинарного типа.

Parameter	type	name	value
Show thumbnail	1	columns.0.show_thumbnail	0 - *(по умолчанию)* Отключено; 1 - Включено.

Расширенная настройка

Следующие параметры расширенной настройки поддерживаются для виджета *Item history*.

Parameter	type	name	value
New values	0	sortorder	0 - *(по умолчанию)* Сверху; 1 - Снизу.

Parameter	type	name	value
Show timestamp	0	show_timestamp	0 - *(по умолчанию)* Отключено; 1 - Включено.
Show column header	0	show_column_header	0 - Выкл.; 1 - Горизонтально; 2 - *(по умолчанию)* Вертикально.
Time period	1	time_period._reference	DASHBOARD._timeperiod - задает селектор периода времени на панели как источник данных; ABCDE._timeperiod - задает совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник данных. По умолчанию: DASHBOARD._timeperiod Кроме того, вы можете задать период времени только в параметрах *From* и *To*.
From	1	time_period.from	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или относительном синтаксисе времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Time period* не задан - *обязателен*, если задан time_period.to
To	1	time_period.to	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или относительном синтаксисе времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Time period* не задан - *обязателен*, если задан time_period.from

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели управления для виджета *Item history*. Дополнительные сведения о настройке панели управления см. в dashboard.create.

Настройка виджета *Item history*

Настройте виджет *Item history*, который отображает последние данные для двух числовых элементов данных "42269" и "42270". Кроме того, настройте отображение столбцов элементов данных по вертикали, а имен столбцов — по горизонтали; ограничьте вывод 15 строками данных и добавьте отдельный столбец с временной меткой.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemhistory",
                        "name": "Item history",
                        "x": "0",
                        "y": "0",
                        "width": "18",
                        "height": "6",
                        "view_mode": "0",
```

```
                    "fields": [
                        {
                            "type": "0",
                            "name": "layout",
                            "value": "1"
                        },
                        {
                            "type": "1",
                            "name": "columns.0.name",
                            "value": "CPU utilization"
                        },
                        {
                            "type": "4",
                            "name": "columns.0.itemid",
                            "value": "42269"
                        },
                        {
                            "type": "1",
                            "name": "columns.1.name",
                            "value": "Memory utilization"
                        },
                        {
                            "type": "4",
                            "name": "columns.1.itemid",
                            "value": "42270"
                        },
                        {
                            "type": "0",
                            "name": "show_lines",
                            "value": "15"
                        },
                        {
                            "type": "0",
                            "name": "show_timestamp",
                            "value": "1"
                        },
                        {
                            "type": "0",
                            "name": "show_column_header",
                            "value": "1"
                        },
                        {
                            "type": "1",
                            "name": "reference",
                            "value": "KIVKD"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
```

```
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

17 Навигатор элементов данных

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Навигатор элементов данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также повышает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Навигатор элементов данных*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Навигатор элементов данных* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	Идентификатор группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета на информационной панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо идентификатора группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.

Параметр	тип	имя	значение
Узлы сети	3	hostids.0	Идентификатор узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета на информационной панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо не настроен, либо настроен как минимум с одной группой узлов сети, к которой относятся настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Узлы сети (виджет/информационная панель)	1	hostids._reference	Вместо идентификатора узла сети: `DASHBOARD.hostid` - задайте выбор узла сети информационной панели в качестве источника данных для узлов сети; `ABCDE._hostid` - задайте совместимый виджет (с параметром *Ссылка*, установленным в "`ABCDE`") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Теги узлов сети			
Тип вычисления	0	host_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ. Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Оператор	0	host_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.

Параметр	тип	имя	значение
Значение тега	1	host_tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на информационной панели шаблона.
Шаблоны элементов данных	1	items.0	Имя или шаблон элемента данных. Примечание: Чтобы настроить несколько шаблонов элементов данных, создайте объект поля виджета на информационной панели для каждого шаблона элемента данных, увеличивая номер в имени свойства.
Теги элементов данных			
Тип вычисления	0	item_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	item_tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Оператор	0	item_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Значение тега	1	item_tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Состояние	0	state	-1 - *(по умолчанию)* Все; 0 - Нормальное; 1 - Не поддерживается.
Показывать проблемы	0	show_problems	0 - Все; 1 - *(по умолчанию)* Не подавленные; 2 - Нет.
Группировать по			

Параметр	тип	имя	значение
Атрибут	0	group_by.0.attribute	0 - Группа узлов сети; 1 - Имя узла сети; 2 - Значение тега узла сети; 3 - Значение тега элемента данных. Примечание: Число в имени свойства указывает порядок атрибута в списке атрибутов группировки. Поведение параметра: - *обязателен* при настройке *Группировать по*
Значение	1	group_by.0.tag_name	Любое строковое значение. Примечание: Число в имени свойства указывает набор атрибутов группировки, заданный в параметре *Атрибут*. Поведение параметра: - *обязателен* при настройке *Группировать по*, если для параметра *Атрибут* задано значение «Значение тега узла сети» или «Значение тега элемента данных»
Ограничение элементов данных	0	show_lines	Допустимые значения находятся в диапазоне от 1 до 9999. По умолчанию: 100.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах информационной панели, которой принадлежит виджет. Поведение параметра: - *обязателен*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Item navigator*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Item navigator*

Настройте виджет *Item navigator*, который отображает до 1000 элементов данных, сгруппированных по узлу сети, а затем по значению тега элемента данных ”component”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemnavigator",
                        "name": "Item navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 0,
```

```
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 1000
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "DFNLK"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update

18 Значение элемента данных

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Значение элемента данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Значение элемента данных*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Item value*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Item	4	itemid.0	ID элемента данных. Поведение параметра: - *required* если *Item (Widget)* не задан
Item (Widget)	1	itemid._reference	Вместо ID элемента данных: `ABCDE._itemid` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для элементов данных. Поведение параметра: - *required* если *Item* не задан
Show	0	show.0	1 - Описание; 2 - Значение; 3 - Время; 4 - Индикатор изменения; 5 - Sparkline. По умолчанию: 1, 2, 3, 4. Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Override host	1	override_hostid._reference	`ABCDE._hostid` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задайте селектор узлов сети панели в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Расширенная настройка

Следующие параметры расширенной настройки поддерживаются для виджета *Значение элемента данных*.

Note:
Число в имени свойства *Thresholds* (например, thresholds.0.color) указывает на позицию порога в списке, отсортированном по возрастанию. Однако если пороги настроены в другом порядке, значения будут отсортированы по возрастанию после обновления конфигурации виджета во веб-интерфейсе Zabbix (например, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Цвет фона	1	bg_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Пороги			
Цвет	1	thresholds.0.color	Шестнадцатеричный код цвета (например, FF0000).
Порог	1	thresholds.0.threshold	Любое строковое значение.
Функция агрегации	0	aggregate_function	0 - *(по умолчанию)* не используется; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` - задать селектор периода времени на панели как источник данных; `ABCDE._timeperiod` - задать совместимый виджет (с параметром `reference`, равным `ABCDE`) как источник данных. По умолчанию: `DASHBOARD._timeperiod` Кроме того, можно задать период времени только в параметрах *From* и *To*. Поведение параметра: - *поддерживается*, если для *Функция агрегации* установлено значение "min", "max", "avg", "count", "sum", "first", "last"
From	1	time_period.from	Допустимая строка времени в абсолютном формате (`YYYY-MM-DD hh:mm:ss`) или относительном синтаксисе времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан и для *Функция агрегации* установлено значение "min", "max", "avg", "count", "sum", "first", "last" - *обязательно*, если задан `time_period.to`
To	1	time_period.to	Допустимая строка времени в абсолютном формате (`YYYY-MM-DD hh:mm:ss`) или относительном синтаксисе времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан и для *Функция агрегации* установлено значение "min", "max", "avg", "count", "sum", "first", "last" - *обязательно*, если задан `time_period.from`
Исторические данные	0	history	0 - *(по умолчанию)* Auto; 1 - History; 2 - Trends.

Описание

Следующие расширенные параметры конфигурации поддерживаются, если для параметра *Показывать* установлено значение «Описание».

Параметр	type	name	value
Описание	1	description	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {ITEM.*}, {INVENTORY.*}, пользовательские макросы. По умолчанию: {ITEM.NAME}.
Горизонтальное положение	0	desc_h_pos	0 - Слева; 1 - *(по умолчанию)* По центру; 2 - Справа. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Вертикальное положение	0	desc_v_pos	0 - Сверху; 1 - Посередине; 2 - *(по умолчанию)* Снизу. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Размер	0	desc_size	Допустимые значения находятся в диапазоне от 1 до 100. По умолчанию: 15.
Полужирный	0	desc_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	desc_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пустое значение).

Значение

Следующие расширенные параметры конфигурации поддерживаются, если для параметра *Отображать* установлено значение «Значение».

Параметр	type	name	value
Десятичные разряды			
Десятичные разряды	0	decimal_places	Допустимые значения: от 1 до 10. По умолчанию: 2.
Размер	0	decimal_size	Допустимые значения: от 1 до 100. По умолчанию: 35.
Положение			
Горизонтальное положение	0	value_h_pos	0 - Слева; 1 - *(по умолчанию)* По центру; 2 - Справа. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Вертикальное положение	0	value_v_pos	0 - Сверху; 1 - *(по умолчанию)* Посередине; 2 - Снизу. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Размер	0	value_size	Допустимые значения: от 1 до 100. По умолчанию: 45.

Параметр	type	name	value
Полужирный шрифт	0	value_bold	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	value_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пустое значение).
Единицы измерения			
Единицы измерения (флажок)	0	units_show	0 - Отключено; 1 - *(по умолчанию)* Включено.
Единицы измерения (значение)	1	units	Любое строковое значение.
Положение	0	units_pos	0 - Перед значением; 1 - Над значением; 2 - *(по умолчанию)* После значения; 3 - Под значением.
Размер	0	units_size	Допустимые значения: от 1 до 100. По умолчанию: 35.
Полужирный шрифт	0	units_bold	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	units_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пустое значение).

Время

Следующие параметры расширенной настройки поддерживаются, если для параметра *Показывать* установлено значение «Время».

Параметр	тип	имя	значение
Горизонтальное положение	0	time_h_pos	0 - Слева; 1 - *(по умолчанию)* По центру; 2 - Справа. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Вертикальное положение	0	time_v_pos	0 - *(по умолчанию)* Сверху; 1 - Посередине; 2 - Снизу. Два или более элемента (Описание, Значение, Время) не могут иметь одинаковые значения параметров *Горизонтальное положение* и *Вертикальное положение*.
Размер	0	time_size	Допустимые значения находятся в диапазоне от 1 до 100. По умолчанию: 15.
Полужирный	0	time_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	time_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пустое значение).

Индикатор изменения

Следующие параметры расширенной настройки поддерживаются, если для *Show* установлено значение ”Change indicator”.

Parameter	type	name	value
Цвет индикатора изменения ↑	1	up_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Цвет индикатора изменения ↓	1	down_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Цвет индикатора изменения ↕	1	updown_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Спарклайн

Поддерживаются следующие расширенные параметры конфигурации, если для параметра *Отображение* установлено значение «Спарклайн».

Параметр	тип	имя	значение
Ширина	0	sparkline.width	Допустимые значения: от 0 до 10. По умолчанию: 1.
Заливка	0	sparkline.fill	Допустимые значения: от 0 до 10. По умолчанию: 3.
Цвет	1	sparkline.color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: 42A5F5.
Период времени	1	sparkline.time_period.reference	DASHBOARD._timeperiod - установить селектор периода времени веб-интерфейса как источник данных; ABCDE._timeperiod - установить совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod Также можно задать период времени только в параметрах *От* и *До*.
От	1	sparkline.time_period.from	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате (now, now/d, now/w-1w и т. д.). Поведение параметра: - *обязателен*, если задан sparkline.time_period.to
До	1	sparkline.time_period.to	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате (now, now/d, now/w-1w и т. д.). Поведение параметра: - *обязателен*, если задан sparkline.time_period.from
Данные истории	0	sparkline.history	0 - *(по умолчанию)* автоматически; 1 - история; 2 - тренды.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Item value*. Дополнительную информацию о настройке панели мониторинга см. в dashboard.create.

Настройка виджета *Item value*

Настройте виджет *Item value*, который отображает значение элемента данных для элемента данных "42266" (доступность агента Zabbix). Кроме того, визуально доработайте виджет с помощью нескольких расширенных параметров, включая динамический цвет фона, который изменяется в зависимости от статуса доступности агента Zabbix.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "item",
                        "name": "Item value",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42266
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "description",
                                "value": "Agent status"
                            },
                            {
                                "type": 0,
                                "name": "desc_h_pos",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "desc_v_pos",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "desc_bold",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "desc_color",
```

```
            "value": "F06291"
        },
        {
            "type": 0,
            "name": "value_h_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "value_size",
            "value": 25
        },
        {
            "type": 1,
            "name": "value_color",
            "value": "FFFF00"
        },
        {
            "type": 0,
            "name": "units_show",
            "value": 0
        },
        {
            "type": 0,
            "name": "time_h_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_v_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_size",
            "value": 10
        },
        {
            "type": 0,
            "name": "time_bold",
            "value": 1
        },
        {
            "type": 1,
            "name": "time_color",
            "value": "9FA8DA"
        },
        {
            "type": 1,
            "name": "thresholds.0.color",
            "value": "E1E1E1"
        },
        {
            "type": 1,
            "name": "thresholds.0.threshold",
            "value": "0"
        },
        {
            "type": 1,
            "name": "thresholds.1.color",
            "value": "D1C4E9"
        },
```

```
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "1"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

19 Карта

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Карта* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карта*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Map*.

Parameter		type	name	value
Refresh interval		0	rf_rate	0 - без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.
Map		8	sysmapid.0	ID карты. Поведение параметра: - *required* если *Map (Widget)* не задан
	Map (Widget)	1	sysmapid._reference	`ABCDE._mapid` - задайте виджет *Map navigation tree* (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для карт. Поведение параметра: - *required* если *Map* не задан
Reference		1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах веб-интерфейса, к которому принадлежит виджет. Поведение параметра: - *required*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Map*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Map*

Настройте виджет *Map*, который отображает карту "1".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 0,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 8,
                                "name": "sysmapid.0",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Настройка связанного виджета *Map*

Настройте виджет *Map*, который связан с виджетом *Map navigation tree*.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 5,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "sysmapid._reference",
                                "value": "ABCDE._mapid"
                            }
                        ]
                    },
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
```

```
                "y": 0,
                "width": 18,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 1,
                        "name": "navtree.1.name",
                        "value": "Element A"
                    },
                    {
                        "type": 1,
                        "name": "navtree.2.name",
                        "value": "Element B"
                    },
                    {
                        "type": 1,
                        "name": "navtree.3.name",
                        "value": "Element C"
                    },
                    {
                        "type": 1,
                        "name": "navtree.4.name",
                        "value": "Element A1"
                    },
                    {
                        "type": 1,
                        "name": "navtree.5.name",
                        "value": "Element A2"
                    },
                    {
                        "type": 1,
                        "name": "navtree.6.name",
                        "value": "Element B1"
                    },
                    {
                        "type": 1,
                        "name": "navtree.7.name",
                        "value": "Element B2"
                    },
                    {
                        "type": 0,
                        "name": "navtree.4.parent",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "navtree.5.parent",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "navtree.6.parent",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "navtree.7.parent",
                        "value": 2
                    },
                    {
                        "type": 0,
```

```
                                "name": "navtree.1.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.2.order",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "navtree.3.order",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "navtree.4.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.5.order",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "navtree.6.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.7.order",
                                "value": 2
                            },
                            {
                                "type": 8,
                                "name": "navtree.6.sysmapid",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "ABCDE"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
```

```
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`
- Дерево навигации карт сетей

20 Дерево навигации по карте

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Дерево навигации по картам* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Дерево навигации по картам*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Map navigation tree*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Не обновлять; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.
Показывать недоступные карты	1	show_unavailable	0 - *(по умолчанию)* Отключено; 1 - Включено.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах веб-интерфейса, к которому принадлежит виджет. Поведение параметра: *- required*

Следующие параметры поддерживаются для настройки элементов дерева навигации по картам.

Parameter	type	name	value
Имя	1	navtree.1.name	Любое строковое значение. Примечание: число в имени свойства задает номер элемента.
Связанная карта	8	navtree.1.sysmapid	ID карты. Примечание: число в имени свойства указывает на элемент, с которым связана карта.
Параметры для создания иерархии элементов	0	navtree.1.parent	Номер родительского элемента. Примечание: число в имени свойства указывает на дочерний элемент. Значение свойства указывает на родительский элемент.
	0	navtree.1.order	Положение элемента в дереве навигации по картам. Примечание: число в имени свойства указывает на номер элемента. Значение свойства указывает на положение элемента в дереве навигации по картам. Положение родительского элемента определяется в пределах всего дерева навигации по картам. Положение дочернего элемента определяется в пределах родительского элемента.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Map navigation tree*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Map navigation tree*

Настройте виджет *Map navigation tree*, который отображает следующее дерево навигации по карте:

- Элемент A
  - Элемент A1
  - Элемент A2
- Элемент B
  - Элемент B1 (содержит связанную карту "1", которую можно отобразить в связанном виджете *Map*)
  - Элемент B2
- Элемент C

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
```

```
            "type": 1,
            "name": "navtree.2.name",
            "value": "Element B"
        },
        {
            "type": 1,
            "name": "navtree.3.name",
            "value": "Element C"
        },
        {
            "type": 1,
            "name": "navtree.4.name",
            "value": "Element A1"
        },
        {
            "type": 1,
            "name": "navtree.5.name",
            "value": "Element A2"
        },
        {
            "type": 1,
            "name": "navtree.6.name",
            "value": "Element B1"
        },
        {
            "type": 1,
            "name": "navtree.7.name",
            "value": "Element B2"
        },
        {
            "type": 0,
            "name": "navtree.4.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.6.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.7.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.1.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.2.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.3.order",
```

```
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "navtree.4.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.5.order",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "navtree.6.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.7.order",
                                "value": 2
                            },
                            {
                                "type": 8,
                                "name": "navtree.6.sysmapid",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "HJQXF"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
```

```
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update
- Карта

21 Круговая диаграмма

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Круговая диаграмма* в методах dashboard.create и dashboard.update.

Attention:
Свойства fields виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Круговая диаграмма*, ознакомьтесь с описанным в таблицах ниже поведением параметров.

Параметры

Для виджета *Pie chart* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.

Набор данных

Для настройки *набора данных* поддерживаются следующие параметры.

Note:
Первое число в имени свойства (например, ds.0.hosts.0, ds.0.items.0) обозначает конкретный набор данных, а второе число, если оно указано, - настроенный узел сети или элемент данных.

Параметр	type	name	value
Тип набора данных	0	ds.0.dataset_type	0 - список элементов данных; 1 - *(по умолчанию)* шаблоны элементов данных.
Элементы данных	4	ds.0.itemids.0	Идентификатор элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне. Примечание: чтобы настроить несколько элементов данных, создайте отдельный объект поля виджета панели для каждого элемента данных, увеличивая число в имени свойства. Поведение параметра: - *обязательный*, если для *типа набора данных* установлено значение «список элементов данных», а параметр *Элементы данных (виджет)* не задан

Параметр	type	name	value
Элементы данных (виджет)	1	ds.0.itemids.0._reference	Вместо идентификатора элемента данных: `ABCDE._itemid` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "ABCDE") в качестве источника данных для элементов данных. Примечание: чтобы настроить несколько виджетов, создайте отдельный объект поля виджета панели для каждого виджета, увеличивая число в имени свойства. Поведение параметра: - *обязательный*, если для *типа набора данных* установлено значение «список элементов данных», а параметр *Элементы данных* не задан
Цвет	1	ds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *обязательный*, если для *типа набора данных* установлено значение «список элементов данных»
Тип элемента данных	0	ds.0.type.0	0 - *(по умолчанию)* обычный; 1 - итоговый. Значение «Итоговый» можно установить только для одного элемента данных во всей диаграмме. Поведение параметра: - *поддерживается*, если для *типа набора данных* установлено значение «список элементов данных»
Шаблоны узлов сети	1	ds.0.hosts.0	Имя или шаблон узла сети (например, "Zabbix*"). Поведение параметра: - *обязательный*, если для *типа набора данных* установлено значение «шаблоны элементов данных» Этот параметр не поддерживается при настройке виджета на панели шаблона.
Шаблоны элементов данных	1	ds.0.items.0	Имя или шаблон элемента данных (например, "*: Number of processed *values per second"). При настройке виджета на панели шаблона следует указывать только шаблоны элементов данных, настроенных в шаблоне. Поведение параметра: - *обязательный*, если для *типа набора данных* установлено значение «шаблоны элементов данных»
Цвет	1	ds.0.color	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если для *типа набора данных* установлено значение «шаблоны элементов данных», а параметр *Цветовая палитра* не задан
Цветовая палитра	0	ds.0.color_palette	Индекс цветовой палитры. Допустимые значения находятся в диапазоне от 0 до 11. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для *типа набора данных* установлено значение «шаблоны элементов данных», а параметр *Цвет* не задан

Параметр	type	name	value
Функция агрегации	0	ds.0.aggregate_function	1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма; 6 - первое значение; 7 - *(по умолчанию)* последнее значение.
Агрегация набора данных	0	ds.0.dataset_aggregation	0 - *(по умолчанию)* отсутствует; 1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма. Поведение параметра: - *поддерживается*, если для *типа элемента данных* установлено значение «итоговый»
Метка набора данных	1	ds.0.data_set_label	Любое строковое значение. По умолчанию: "" (пустое значение).
Теги элементов данных			
Тип проверки	0	ds.0.item_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	ds.0.item_tags.0.tag	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке проверки тегов. Поведение параметра: - *обязательный* при настройке *тегов элементов данных*
Оператор	0	ds.0.item_tags.0.operator	0 - содержит; 1 - равно; 2 - не содержит; 3 - не равно; 4 - существует; 5 - не существует. Примечание: число в имени свойства указывает порядок тега в списке проверки тегов. Поведение параметра: - *обязательный* при настройке *тегов элементов данных*
Значение тега	1	ds.0.item_tags.0.value	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке проверки тегов. Поведение параметра: - *обязательный* при настройке *тегов элементов данных*

Параметры отображения

Для настройки *параметров отображения* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Выбор данных истории	0	source	0 - *(по умолчанию)* Автоматически; 1 - История; 2 - Тренды.

Параметр	тип	имя	значение
Отображение	0	draw_type	0 - *(по умолчанию)* Круговая диаграмма; 1 - Кольцевая диаграмма.
Ширина	0	width	20 - 20% радиуса; 30 - 30% радиуса; 40 - 40% радиуса; 50 - *(по умолчанию)* 50% радиуса. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Кольцевая диаграмма»
Ширина обводки	0	stroke	Допустимые значения: от 0 до 10. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Кольцевая диаграмма»
Показывать общее значение	0	total_show	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Отображение* установлено значение «Кольцевая диаграмма»
Размер	0	value_size_type	0 - *(по умолчанию)* Автоматически; 1 - Пользовательский. Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»
Размер (значение пользовательского размера)	0	value_size	Допустимые значения: от 1 до 100. По умолчанию: 20. Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»
Десятичные разряды	0	decimal_places	Допустимые значения: от 0 до 6. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»
Единицы измерения (флажок)	0	units_show	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»
Единицы измерения (значение)	1	units	Любое строковое значение. Поведение параметра: - *поддерживается*, если для параметра *Единицы измерения* (флажок) установлено значение «Включено»
Полужирный шрифт	0	value_bold	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»

Параметр	тип	имя	значение
Цвет	1	value_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для параметра *Показывать общее значение* установлено значение «Включено»
Расстояние между секторами	0	space	Допустимые значения: от 0 до 10. По умолчанию: 1.
Объединять сектора размером менее N% (флажок)	0	merge	0 - *(по умолчанию)* Отключено; 1 - Включено.
Объединять сектора размером менее N% (значение)	0	merge_percent	Допустимые значения: от 1 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для параметра *Объединять сектора размером менее N%* (флажок) установлено значение «Включено»
Объединять сектора размером менее N% (цвет)	1	merge_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для параметра *Объединять сектора размером менее N%* (флажок) установлено значение «Включено»

Период времени

Следующие параметры поддерживаются для настройки *Периода времени*.

Parameter	type	name	value
Период времени	1	time_period._reference	DASHBOARD._timeperiod - задать селектор периода времени на веб-интерфейсе панели как источник данных; ABCDE._timeperiod - задать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник данных. По умолчанию: DASHBOARD._timeperiod Кроме того, вы можете задать период времени только в параметрах *From* и *To*.
From	1	time_period.from	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *supported*, если *Период времени* не задан - *required*, если задан time_period.to
To	1	time_period.to	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *supported*, если *Период времени* не задан - *required*, если задан time_period.from

Легенда

Для настройки *Легенды* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Показывать легенду	0	legend	0 - Отключено; 1 - *(по умолчанию)* Включено.
Показывать значение	0	legend_value	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Показывать функцию агрегации	0	legend_aggregation	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Строки	0	legend_lines_mode	0 - *(по умолчанию)* Фиксированное количество; 1 - Переменное количество. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Количество строк/ Максимальное количество строк	0	legend_lines	Допустимые значения: от 1 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено»
Количество столбцов	0	legend_columns	Допустимые значения: от 1 до 4. По умолчанию: 4. Поведение параметра: - *поддерживается*, если для параметра *Показывать легенду* установлено значение «Включено», а для параметра *Показывать значение* установлено значение «Отключено»

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Pie chart*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *круговая диаграмма*

Настройте виджет *круговая диаграмма* следующим образом:

- 2 набора данных с общим количеством 9 элементов данных на 1 узле сети.
- Первый набор данных имеет тип ”Item list” и состоит из 3 элементов данных, все с типом ”Normal” и отображаются разными цветами.
- Второй набор данных имеет тип ”Item patterns”, состоит из 6 элементов данных, для каждого элемента данных настроена агрегация и отображается пользовательским цветом.
- Второй набор данных также имеет пользовательскую метку набора данных.
- Данные на круговой диаграмме отображаются как кольцевая диаграмма с пользовательской шириной и общим значением с единицами измерения в центре.
- Данные на круговой диаграмме отображаются и агрегируются за пользовательский период времени — последние 3 часа.
- Легенда круговой диаграммы отображает настроенные элементы в 4 строках.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
```

```
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
                        "y": 0,
                        "width": 24,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
                                "type": 0,
                                "name": "ds.0.type.0",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23269
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "BF00FF"
                            },
                            {
                              "type": 0,
                              "name": "ds.0.type.1",
                              "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23257
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.2",
                                "value": "0040FF"
                            },
                            {
```

```
                    "type": 0,
                    "name": "ds.0.type.2",
                    "value": 0
                },
                {
                    "type": 1,
                    "name": "ds.1.hosts.0",
                    "value": "Zabbix server"
                },
                {
                    "type": 1,
                    "name": "ds.1.items.0",
                    "value": "*: Number of processed *values per second"
                },
                {
                    "type": 1,
                    "name": "ds.1.color",
                    "value": "000000"
                },
                {
                    "type": 0,
                    "name": "ds.1.aggregate_function",
                    "value": 3
                },
                {
                    "type": 1,
                    "name": "ds.1.data_set_label",
                    "value": "Number of processed values per second"
                },
                {
                    "type": 0,
                    "name": "draw_type",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "width",
                    "value": 30
                },
                {
                    "type": 0,
                    "name": "total_show",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "units_show",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "graph_time",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "time_period.from",
                    "value": "now-3h"
                },
                {
                    "type": 1,
                    "name": "time_period.to",
```

```
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

22 проблемных узла сети

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Проблемные узлы сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Проблемные узлы сети*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Проблемные узлы сети*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Группы узлов сети (виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Исключить группы узлов сети	2	exclude_groupids.0	ID группы узлов сети. Примечание: чтобы исключить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Узлы сети	3	hostids.0	ID узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо вообще не настроен, либо настроен как минимум с одной группой узлов сети, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Узлы сети (виджет/панель)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Проблема	1	problem	Имя события проблемы (без учета регистра, полное имя или его часть).

Parameter	type	name	value
Серьезность	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Теги проблемы			
Тип оценки	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраиваются *Теги проблемы*
Показывать подавленные проблемы	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Скрывать группы без проблем	0	hide_empty_groups	0 - *(по умолчанию)* Отключено; 1 - Включено. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Отображение проблем	0	ext_ack	0 - *(по умолчанию)* Все; 1 - Только непризнанные; 2 - Раздельно.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, к которой принадлежит виджет. Поведение параметра: - *обязательно*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Проблемные узлы сети*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Problem hosts*

Настройте виджет *Problem hosts*, который отображает узлы сети из групп узлов сети "2" и "4", у которых есть проблемы с названием, содержащим строку "CPU", и которые имеют следующие уровни серьезности: "Warning", "Average", "High", "Disaster".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemhosts",
                        "name": "Problem hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "problem",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
```

```
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

23 проблемы

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Проблемы* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Проблемы*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Проблемы*.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Показывать	0	show	1 - *(по умолчанию)* Последние проблемы; 2 - История; 3 - Проблемы.
Группы узлов сети	2	groupids.0	Идентификатор группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели мониторинга для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо идентификатора группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Исключить группы узлов сети	2	exclude_groupids.0	Идентификатор группы узлов сети. Примечание: Чтобы исключить несколько групп узлов сети, создайте объект поля виджета панели мониторинга для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Узлы сети	3	hostids.0	Идентификатор узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета панели мониторинга для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо не настроен, либо настроен как минимум с одной группой узлов сети, к которой относятся настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Узлы сети (виджет/панель мониторинга)	1	hostids._reference	Вместо идентификатора узла сети: `DASHBOARD.hostids` - задайте выбор узлов сети панели мониторинга в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Проблема	1	problem	Имя события проблемы (без учета регистра, полное имя или его часть).

Параметр	тип	имя	значение
Важность	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели мониторинга для каждого значения, увеличивая номер в имени свойства.
Теги проблем			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблем*
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблем*
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблем*
Показывать теги	0	show_tags	0 - *(по умолчанию)* Нет; 1 - 1; 2 - 2; 3 - 3.
Имя тега (формат)	0	tag_name_format	0 - *(по умолчанию)* Полное; 1 - Сокращенное; 2 - Нет. Поведение параметра: - *поддерживается*, если для параметра *Показывать теги* установлено значение ”1”, ”2” или ”3”
Приоритет отображения тегов	1	tag_priority	Список тегов, разделенных запятыми. Поведение параметра: - *поддерживается*, если для параметра *Показывать теги* установлено значение ”1”, ”2” или ”3”
Показывать операционные данные	0	show_opdata	0 - *(по умолчанию)* Нет; 1 - Отдельно; 2 - С именем проблемы.

Параметр	тип	имя	значение
Показывать подавленные проблемы	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Статус подтверждения	0	acknowledgement_status	0 - *(по умолчанию)* Все; 1 - Не подтверждено; 2 - Подтверждено.
Мною	0	acknowledged_by_me	0 - *(по умолчанию)* Отключено; 1 - Включено.
Сортировать записи по	0	sort_triggers	1 - Важность (по убыванию); 2 - Узел сети (по возрастанию); 3 - Время (по возрастанию); 4 - *(по умолчанию)* Время (по убыванию); 13 - Важность (по возрастанию); 14 - Узел сети (по убыванию); 15 - Проблема (по возрастанию); 16 - Проблема (по убыванию). Для всех значений, кроме ”Время (по убыванию)” и ”Время (по возрастанию)”, параметр *Показывать временную шкалу* должен иметь значение ”Отключено”. Значения ”Узел сети (по возрастанию)” и ”Узел сети (по убыванию)” не поддерживаются при настройке виджета на панели мониторинга шаблона.
Показывать временную шкалу	0	show_timeline	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если для параметра *Сортировать записи по* установлено значение ”Время (по убыванию)” или ”Время (по возрастанию)”
Выделять всю строку	0	highlight_row	0 - *(по умолчанию)* Отключено; 1 - Включено. Этот параметр не поддерживается в темах с высокой контрастностью.
Показывать строки	0	show_lines	Допустимый диапазон значений: от 1 до 100. По умолчанию: 25.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели мониторинга, которой принадлежит виджет. Поведение параметра: - *обязательна*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Problems*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Проблемы*

Настройте виджет *Проблемы*, который отображает проблемы для группы узлов сети ”4”, удовлетворяющие следующим условиям:

- Проблемы, у которых есть тег с именем ”scope”, содержащий значения ”performance” или ”availability”, или ”capacity”.
- Проблемы со следующими уровнями серьезности: ”Warning”, ”Average”, ”High”, ”Disaster”.

Кроме того, настройте виджет так, чтобы он показывал теги и оперативные данные.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "name": "Problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "performance"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "availability"
                            },
                            {
                                "type": 1,
                                "name": "tags.2.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.2.operator",
```

```
                            "value": 0
                        },
                        {
                            "type": 1,
                            "name": "tags.2.value",
                            "value": "capacity"
                        },
                        {
                            "type": 0,
                            "name": "severities.0",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "severities.1",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "severities.2",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "severities.3",
                            "value": 5
                        },
                        {
                            "type": 0,
                            "name": "show_tags",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "show_opdata",
                            "value": 1
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

24 Проблемы по уровню серьезности

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Проблемы по степени серьезности* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Проблемы по степени серьезности*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Problems by severity*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Host groups	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Host groups (Widget)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Exclude host groups	2	exclude_groupids.0	ID группы узлов сети. Примечание: чтобы исключить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Parameter	type	name	value
Hosts	3	hostids.0	ID узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Host groups* должен либо вообще не быть настроен, либо быть настроен как минимум с одной группой узлов сети, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Hosts (Widget/Dashboard)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Problem	1	problem	Имя события проблемы (без учета регистра, полное имя или его часть).
Severity	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Problem tags			
Evaluation type	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Tag name	1	tags.0.tag	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Problem tags*
Operator	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Problem tags*

Parameter	type	name	value
Tag value	1	tags.0.value	Любое строковое значение. Примечание: номер в имени свойства соответствует порядку тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Problem tags*
Show	0	show_type	0 - *(по умолчанию)* Группы узлов сети; 1 - Итоги. Этот параметр не поддерживается при настройке виджета на шаблонной панели и по умолчанию установлен в ”Итоги”.
Layout	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный. Поведение параметра: - *supported* если *Show* установлен в ”Totals”
Show operational data	0	show_opdata	0 - *(по умолчанию)* Нет; 1 - Отдельно; 2 - С именем проблемы.
Show suppressed problems	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Hide groups without problems	0	hide_empty_groups	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *supported* если *Show* установлен в ”Host groups” Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Problem display	0	ext_ack	0 - *(по умолчанию)* Все; 1 - Только не подтвержденные; 2 - Раздельно.
Show timeline	0	show_timeline	0 - Отключено; 1 - *(по умолчанию)* Включено.
Reference	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, которой принадлежит виджет. Поведение параметра: - *required*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Проблемы по степени серьезности*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Проблемы по степени серьезности*

Настройте виджет *Проблемы по степени серьезности*, который отображает общее количество проблем для всех групп узлов сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemsbysv",
                        "name": "Problems by severity",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_type",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update

25 Точечная диаграмма

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Точечная диаграмма* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Точечная диаграмма*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Точечная диаграмма* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, к которой принадлежит виджет. Поведение параметра: - *обязательный*

Набор данных

Для настройки *набора данных* поддерживаются следующие параметры.

Note:
Первое число в имени свойства (например, ds.0.hosts.0, ds.0.items.0) обозначает конкретный набор данных, а второе число, если оно присутствует, обозначает настроенный узел сети или элемент данных.

Параметр	type	name	value
Тип набора данных ***Набор данных: Список элементов данных***	0	ds.0.dataset_type	0 - Список элементов данных; 1 - *(по умолчанию)* Шаблоны элементов данных.
Цвет	1	ds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных"

Параметр	type	name	value
Ось X	4	ds.0.x_axis_itemids.0	ID элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне. Примечание: Чтобы настроить несколько элементов данных, создайте объект поля виджета панели для каждого элемента данных с увеличенным номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Ось X (Виджет)* не задана
Ось X (Виджет)	1	ds.0.x_axis_itemids.0.reference	ID элемента данных: `ABCDE._itemid` - задать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для элементов данных. Примечание: Чтобы настроить несколько виджетов, создайте объект поля виджета панели для каждого виджета с увеличенным номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Ось X* не задана
Ось Y	4	ds.0.y_axis_itemids.0	ID элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне. Примечание: Чтобы настроить несколько элементов данных, создайте объект поля виджета панели для каждого элемента данных с увеличенным номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Ось Y (Виджет)* не задана
Ось Y (Виджет)	1	ds.0.y_axis_itemids.0.reference	ID элемента данных: `ABCDE._itemid` - задать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для элементов данных. Примечание: Чтобы настроить несколько виджетов, создайте объект поля виджета панели для каждого виджета с увеличенным номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Ось Y* не задана
Переопределить узел сети	1	ds.0.override_hostid.0.reference	`ABCDE._hostid`- задать совместимый виджет (с параметром "Reference", установленным в "ABCDE") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задать селектор узла сети панели в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Набор данных: Шаблоны элементов данных			

Параметр	type	name	value
Цвет	1	ds.0.color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных” и *Палитра цветов* не задана
Палитра цветов	0	ds.0.color_palette	Индекс палитры цветов. Возможный диапазон значений: 0-11. По умолчанию: 0. Поведение параметра: - *поддерживается*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных” и *Цвет* не задан
Шаблоны узлов сети	1	ds.0.hosts.0	Имя узла сети или шаблон (например, ”Zabbix*”). Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных” Этот параметр не поддерживается при настройке виджета на панели шаблона.
Шаблоны элементов данных оси X	1	ds.0.x_axis_items.0	Имя элемента данных или шаблон (например, ”*: Number of processed *values per second”). При настройке виджета на панели шаблона следует указывать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных”
Шаблоны элементов данных оси Y	1	ds.0.y_axis_items.0	Имя элемента данных или шаблон (например, ”*: Number of processed *values per second”). При настройке виджета на панели шаблона следует указывать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных”
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети с увеличенным номером в имени свойства. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Группы узлов сети (Виджет)	1	groupids._reference	Вместо ID группы узлов сети: ABCDE._hostgroupids - задать совместимый виджет (с параметром *Reference*, установленным в ”ABCDE”) в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
Переопределить узел сети	1	ds.0.override_hostid.ABCDE.reference	`ABCDE._hostid`- задать совместимый виджет (с параметром "Reference", установленным в "ABCDE") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задать селектор узла сети панели в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги узлов сети			
Тип вычисления	0	evaltype_host	0 - *(по умолчанию)* And/Or; 2 - Or. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Оператор	0	host_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Значение тега	1	host_tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Маркер	0	ds.0.marker	0 - *(по умолчанию)* Эллипс; 1 - Квадрат; 2 - Треугольник; 3 - Ромб; 4 - Звезда; 5 - Крест.
Размер маркера	0	ds.0.marker_size	0 - *(по умолчанию)* Маленький (6px); 1 - Средний (9px); 2 - Большой (15px).

Параметр	type	name	value
Сдвиг времени	1	ds.0.timeshift	Допустимая строка времени (например, 3600, 1h и т. д.). Можно использовать суффиксы времени. Также допускаются отрицательные значения. По умолчанию: "" (пусто).
Интервал агрегации	1	ds.0.aggregate_interval	Допустимая строка времени (например, 3600, 1h и т. д.). Можно использовать суффиксы времени. По умолчанию: 15m.
Функция агрегации	0	ds.0.aggregate_function	0 - не используется; 1 - min; 2 - max; 3 - *(по умолчанию)* avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Параметры отображения

Для настройки *Параметров отображения* поддерживаются следующие параметры.

Параметр	type	name	value
Выбор исторических данных	0	source	0 - *(по умолчанию)* Авто; 1 - История; 2 - Тренды.
Имена узлов сети в метках	0	show_hostnames	0 - *(по умолчанию)* Авто; 1 - Показывать; 2 - Скрывать. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Период времени

Для настройки *Периода времени* поддерживаются следующие параметры.

Параметр	type	name	value
Период времени	1	time_period._reference	DASHBOARD._timeperiod — установить селектор периода времени панели мониторинга в качестве источника данных; ABCDE._timeperiod — установить совместимый виджет (с его параметром *Reference*, установленным в "ABCDE") в качестве источника данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы можно задать период времени только в параметрах *От* и *До*.
От	1	time_period.from	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или в относительном синтаксисе времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан time_period.to

Параметр	type	name	value
До	1	time_period.to	Допустимая строка времени в абсолютном формате (`YYYY-MM-DD hh:mm:ss`) или в относительном синтаксисе времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан `time_period.from`

Оси

Для настройки *осей* поддерживаются следующие параметры.

Параметр	type	name	value
Ось X	0	x_axis	0 - Отключено; 1 - *(по умолчанию)* Включено.
Ось Y	0	y_axis	0 - *(по умолчанию)* Отключено; 1 - Включено.
Мин.	1	x_axis_min	Любое числовое значение. По умолчанию: "" (пусто).
		y_axis_min	
Макс.	1	x_axis_max	Любое числовое значение. По умолчанию: "" (пусто).
		y_axis_max	
Единицы измерения (тип)	0	x_axis_units	0 - *(по умолчанию)* Авто; 1 - Статически.
		y_axis_units	
Единицы измерения (значение)	1	x_axis_static_units	Любое строковое значение. По умолчанию: "" (пусто).
		y_axis_static_units	

Легенда

Следующие параметры поддерживаются для настройки *легенды*.

Параметр	type	name	value
Показывать легенду	0	legend	0 - Отключено; 1 - *(по умолчанию)* Включено.
Показывать функцию агрегации	0	legend_aggregation	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Показывать легенду* установлено в ”Включено”
Строки	0	legend_lines_mode	0 - *(по умолчанию)* Фиксировано; 1 - Переменно. Поведение параметра: - *поддерживается*, если *Показывать легенду* установлено в ”Включено”
Количество строк/ Максимальное количество строк	0	legend_lines	Возможный диапазон значений: 1-10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если *Показывать легенду* установлено в ”Включено”

Параметр	type	name	value
Количество столбцов	0	legend_columns	Возможный диапазон значений: 1-4. По умолчанию: 4. Поведение параметра: - *поддерживается*, если *Показывать легенду* установлено в "Включено"

Пороговые значения

Для настройки *пороговых значений* поддерживаются следующие параметры.

Note:
Число в имени свойства *Thresholds* (например, thresholds.0.color) указывает на позицию порогового значения в списке, отсортированном по возрастанию. Однако, если пороговые значения настроены в другом порядке, после обновления конфигурации виджета в веб-интерфейсе Zabbix значения будут отсортированы по возрастанию (например, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Параметр	type	name	value
Интерполяция цвета	0	interpolation	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	thresholds.0.color	Шестнадцатеричный код цвета (например, FF0000).
Ось X	1	thresholds.0.x_axis_threshold	Любое числовое значение. Поддерживаются суффиксы (например, "1d", "2w", "4K", "8G").
Ось Y	1	thresholds.0.y_axis_threshold	Любое числовое значение. Поддерживаются суффиксы (например, "1d", "2w", "4K", "8G").

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Scatter plot*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Scatter plot*

Настройте виджет *Scatter plot* следующим образом:

- 5 наборов данных типа "Шаблоны элементов данных", каждый с агрегированием по умолчанию (avg, 15m) и разными маркерами.
- Легенда отображает настроенные элементы данных в одной строке и двух столбцах.
- Пороговые значения настроены так, чтобы окрашивать маркер в красный цвет, когда его значение достигает 80 по любой из осей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "scatterplot",
                        "name": "Scatter plot",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
```

```
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 1,
                            "name": "ds.0.color",
                            "value": "0C5A87"
                        },
                        {
                            "type": 1,
                            "name": "ds.0.hosts.0",
                            "value": "de-frankfurt*"
                        },
                        {
                            "type": 1,
                            "name": "ds.0.x_axis_items.0",
                            "value": "Memory utilization"
                        },
                        {
                            "type": 1,
                            "name": "ds.0.y_axis_items.0",
                            "value": "CPU utilization"
                        },
                        {
                            "type": 0,
                            "name": "ds.0.marker",
                            "value": 0
                        },
                        {
                            "type": 0,
                            "name": "ds.0.marker_size",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "ds.1.color",
                            "value": "66B0D9"
                        },
                        {
                            "type": 1,
                            "name": "ds.1.hosts.0",
                            "value": "fr-paris*"
                        },
                        {
                            "type": 1,
                            "name": "ds.1.x_axis_items.0",
                            "value": "Memory utilization"
                        },
                        {
                            "type": 1,
                            "name": "ds.1.y_axis_items.0",
                            "value": "CPU utilization"
                        },
                        {
                            "type": 0,
                            "name": "ds.1.marker",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "ds.1.marker_size",
                            "value": 1
                        },
```

```
                    {
                        "type": 1,
                        "name": "ds.2.color",
                        "value": "0A466A"
                    },
                    {
                        "type": 1,
                        "name": "ds.2.hosts.0",
                        "value": "lv-riga*"
                    },
                    {
                        "type": 1,
                        "name": "ds.2.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.2.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.2.marker",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "ds.2.marker_size",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.3.color",
                        "value": "3394C3"
                    },
                    {
                        "type": 1,
                        "name": "ds.3.hosts.0",
                        "value": "pl-warsaw*"
                    },
                    {
                        "type": 1,
                        "name": "ds.3.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.3.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.3.marker",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "ds.3.marker_size",
                        "value": 1
                    },
                    {
                        "type": 1,
```

```
            "name": "ds.4.color",
            "value": "1492C8"
        },
        {
            "type": 1,
            "name": "ds.4.hosts.0",
            "value": "se-stockholm*"
        },
        {
            "type": 1,
            "name": "ds.4.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.4.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.4.marker",
            "value": 4
        },
        {
            "type": 0,
            "name": "ds.4.marker_size",
            "value": 1
        },
        {
            "type": 0,
            "name": "legend_columns",
            "value": 2
        },
        {
            "type": 0,
            "name": "interpolation",
            "value": 1
        },
        {
            "type": 1,
            "name": "thresholds.0.color",
            "value": "D40000"
        },
        {
            "type": 1,
            "name": "thresholds.0.x_axis_threshold",
            "value": "80"
        },
        {
            "type": 1,
            "name": "thresholds.0.y_axis_threshold",
            "value": ""
        },
        {
            "type": 1,
            "name": "thresholds.1.color",
            "value": "D40000"
        },
        {
            "type": 1,
            "name": "thresholds.1.x_axis_threshold",
            "value": ""
```

```
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.y_axis_threshold",
                                "value": "80"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "JQISY"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

См. также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

26 Отчет SLA

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Отчёт SLA* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создаёт риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Отчёт SLA*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Отчет SLA* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - *(по умолчанию)* Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
SLA	10	slaid.0	Идентификатор SLA. Поведение параметра: - *обязательный*
Сервис	9	serviceid.0	Идентификатор сервиса.
Показывать периоды	0	show_periods	Допустимые значения: от 1 до 100. По умолчанию: 20.
От	1	date_from	Допустимая строка даты в формате `YYYY-MM-DD`. Поддерживаются относительные даты с модификаторами `d`, `w`, `M`, `y` (например, `now`, `now/d`, `now/w-1w` и т. д.).
До	1	date_to	Допустимая строка даты в формате `YYYY-MM-DD`. Поддерживаются относительные даты с модификаторами `d`, `w`, `M`, `y` (например, `now`, `now/d`, `now/w-1w` и т. д.).

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *SLA report*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *отчета SLA*

Настройте виджет *отчета SLA*, который отображает отчет SLA для SLA "4" и сервиса "2" за период последних 30 дней.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "slareport",
                        "name": "SLA report",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 10,
                                "name": "slaid.0",
                                "value": 4
                            },
                            {
                                "type": 9,
                                "name": "serviceid.0",
```

```
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "date_from",
                                "value": "now-30d"
                            },
                            {
                                "type": 1,
                                "name": "date_to",
                                "value": "now"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- dashboard.create
- dashboard.update

27 Системная информация

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Системная информация* в методах dashboard.create и dashboard.update.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Системная информация*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *System Information*.

Parameter	type	name	value
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - *(по умолчанию)* 15 минут.
Показывать	0	info_type	0 - *(по умолчанию)* Статистика системы; 1 - Узлы высокой доступности.
Показывать подробности проверки обновлений ПО	0	show_software_update_check_details	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *supported* если `AllowSoftwareUpdateCheck` в конфигурации сервера Zabbix включен и для параметра *Показывать* установлено значение "System stats"

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *System information*. Дополнительные сведения о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *System information*

Настройте виджет *System information*, который отображает системную статистику с интервалом обновления 10 минут и включенной проверкой обновлений программного обеспечения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "systeminfo",
                        "name": "System information",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            },
```

```
                            {
                                "type": 0,
                                "name": "show_software_update_check_details",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

28 Лучшие узлы сети

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Топ узлов сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Топ узлов сети*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Лучшие узлы сети* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Узлы сети	3	hostids.0	ID узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо не настроен, либо настроен как минимум с одной группой узлов сети, к которой относятся настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Узлы сети (виджет/панель)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте выбор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Ссылка*, установленным в значение "`ABCDE`") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги узлов сети			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	тип	имя	значение
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Показывать узлы сети на обслуживании	0	maintenance	0 - *(по умолчанию)* Отключено; 1 - Включено.
Столбцы (см. ниже)			
Сортировать по	0	column	Числовое значение столбца из настроенных столбцов.
Порядок	0	order	2 - *(по умолчанию)* Первые N; 3 - Последние N.
Ограничение узлов сети	0	show_lines	Допустимые значения: от 1 до 1000. По умолчанию: 10. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Столбцы

У столбцов есть общие параметры и дополнительные параметры в зависимости от настройки параметра *Data*.

Note:
Для всех параметров, связанных со столбцами, число в имени свойства (например, columns.0.name) указывает на столбец, для которого настроен параметр.

Следующие параметры поддерживаются для всех столбцов.

Parameter	type	name	value
Имя	1	columns.0.name	Любое строковое значение. Поведение параметра: - *required*

Parameter	type	name	value
Данные	0	columns.0.data	1 - значение элемента данных; 2 - имя узла сети; 3 - текст. Поведение параметра: - *required*
Основной цвет	1	columns.0.base_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *required*

Значение элемента данных

Следующие параметры поддерживаются, если для *Data* установлено значение «Item value».

Note:
Первое число в имени свойства *Thresholds* (например, columnsthresholds.0.color.0) указывает на столбец, для которого настроены пороговые значения, а второе число указывает позицию порогового значения в списке, отсортированном по возрастанию. Однако если пороговые значения настроены в другом порядке, после обновления конфигурации виджета в веб-интерфейсе Zabbix значения будут отсортированы по возрастанию (например, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Параметр	type	name	value
Элемент данных	1	columns.0.item	Допустимое имя элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне.
Отображать значение элемента данных как	0	columns.0.display_value_as	0 - *(по умолчанию)* Числовое значение; 1 - Текст; 2 - Двоичное значение.
Отображение	0	columns.0.display	1 - *(по умолчанию)* Как есть; 2 - Гистограмма; 3 - Индикаторы; 6 - Спарклайн. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric»
Минимум	1	columns.0.min	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Bar» или «Indicators»
Максимум	1	columns.0.max	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Bar» или «Indicators»
Десятичные разряды	0	columns.0.decimal_places	Допустимы значения от 0 до 10. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric»

Параметр	type	name	value
Спарклайн			
Ширина	0	columns.0.sparkline.width	Допустимы значения от 0 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline»
Заливка	0	columns.0.sparkline.fill	Допустимы значения от 0 до 10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline»
Цвет	1	columns.0.sparkline.color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: `42A5F5`. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline»
Период времени	1	columns.0.sparkline.time_period._timeperiod	`DASHBOARD._timeperiod` - использовать селектор периода времени панели мониторинга как источник данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) как источник данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline»
От	1	columns.0.sparkline.time_period.from	Поддерживаются строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline» - *требуется*, если задан `columns.0.sparkline.time_period.to`
До	1	columns.0.sparkline.time_period.to	Поддерживаются строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline» - *требуется*, если задан `columns.0.sparkline.time_period.from`

Параметр	type	name	value
Данные истории	0	columns.0.sparkline.history	0 - *(по умолчанию)* Автоматически; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric», а для *Отображение* — «Sparkline»
Пороговые значения			
Цвет	1	columnsthresholds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric»
Пороговое значение	1	columnsthresholds.0.threshold.0	Любое строковое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric»
Выделения			
Цвет	1	columns.0.highlights.0.color	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Text»
Шаблон	1	columns.0.highlights.0.pattern	Любое строковое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Text»
Показывать миниатюру	0	columns.0.show_thumbnail	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Binary»
Функция агрегации	0	columns.0.aggregate_function	0 - *(по умолчанию)* не используется; 1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма; 6 - первое значение; 7 - последнее значение.
Период времени	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - использовать селектор периода времени панели мониторинга как источник данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) как источник данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если для *Функция агрегации* установлено значение «min», «max», «avg», «count», «sum», «first» или «last»

Параметр	type	name	value
От	1	columns.0.time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан, а для *Функция агрегации* установлено значение «min», «max», «avg», «count», «sum», «first» или «last» - *требуется*, если задан `columns.0.time_period.to`
До	1	columns.0.time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан, а для *Функция агрегации* установлено значение «min», «max», «avg», «count», «sum», «first» или «last» - *требуется*, если задан `columns.0.time_period.from`
Данные истории	0	columns.0.history	0 - *(по умолчанию)* Автоматически; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение «Numeric»
Ссылка	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели мониторинга, которой принадлежит виджет. Поведение параметра: - *требуется*

Текст

Следующие параметры поддерживаются, если для *Data* установлено значение ”Text”.

Parameter	type	name	value
Text	1	columns.0.text	Любое строковое значение, включая макросы. Поддерживаемые макросы: `{HOST.*}`, `{INVENTORY.*}`. Поведение параметра: - *required*, если для *Data* установлено значение ”Text”

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета информационной панели для виджета *Топ узлов сети*. Для получения дополнительной информации о настройке информационной панели см. `dashboard.create`.

Настройка виджета *Top hosts*

Настройте виджет *Top hosts*, который отображает узлы сети с наибольшей загрузкой CPU в группе узлов сети ”4”. Кроме того, настройте следующие пользовательские столбцы: ”Имя узла сети”, ”Загрузка CPU в %”, ”среднее за 1 мин”, ”среднее за 5 мин”, ”среднее за 15 мин”, ”Процессы”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
```

```
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "tophosts",
                        "name": "Top hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "columns.0.name",
                                "value": "Host"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.data",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "columns.0.base_color",
                                "value": "FFFFFF"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.name",
                                "value": "CPU utilization in %"
                            },
                            {
                                "type": 0,
                                "name": "columns.1.data",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "columns.1.base_color",
                                "value": "4CAF50"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.item",
                                "value": "CPU utilization"
                            },
                            {
                                "type": 0,
                                "name": "columns.1.display",
                                "value": 3
                            },
                            {
                                "type": 1,
```

```
            "name": "columns.1.min",
            "value": "0"
        },
        {
            "type": 1,
            "name": "columns.1.max",
            "value": "100"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.0",
            "value": "FFFF00"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.0",
            "value": "50"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.1",
            "value": "FF8000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.1",
            "value": "80"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.2",
            "value": "FF4000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.2",
            "value": "90"
        },
        {
            "type": 1,
            "name": "columns.2.name",
            "value": "1m avg"
        },
        {
            "type": 0,
            "name": "columns.2.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.2.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.2.item",
            "value": "Load average (1m avg)"
        },
        {
            "type": 1,
            "name": "columns.3.name",
            "value": "5m avg"
```

```
        },
        {
            "type": 0,
            "name": "columns.3.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.3.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.3.item",
            "value": "Load average (5m avg)"
        },
        {
            "type": 1,
            "name": "columns.4.name",
            "value": "15m avg"
        },
        {
            "type": 0,
            "name": "columns.4.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.4.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.4.item",
            "value": "Load average (15m avg)"
        },
        {
            "type": 1,
            "name": "columns.5.name",
            "value": "Processes"
        },
        {
            "type": 0,
            "name": "columns.5.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.5.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.5.item",
            "value": "Number of processes"
        },
        {
            "type": 0,
            "name": "columns.5.decimal_places",
            "value": 0
        },
        {
```

```
                                "type": 0,
                                "name": "column",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

29 Элементы данных верхнего уровня

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Топ элементов данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Топ элементов данных*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Top items* поддерживаются следующие параметры.

Параметр	type	name	value
Интервал обновления	0	rf_rate	0 - Не обновлять; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждой группы узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Группы узлов сети (Виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` — укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Узлы сети	3	hostids.0	ID узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого узла сети. Для нескольких узлов сети параметр *Группы узлов сети* либо не должен быть настроен вовсе, либо должен быть настроен как минимум с одной группой узлов сети, которой принадлежат указанные узлы сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Узлы сети (Виджет/Панель)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` — укажите селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` — укажите совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги узлов сети			
Тип вычисления	0	host_tags_evaltype	0 - *(по умолчанию)* И/Или; 2 - Или. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: Номер в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
Оператор	0	host_tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: Номер в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Значение тега	1	host_tags.0.value	Любое строковое значение. Примечание: Номер в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный.
Показывать проблемы	0	show_problems	0 - Все; 1 - *(по умолчанию)* Неподавленные; 2 - Нет.
Элементы данных (см. ниже)			

Расширенная настройка

Для виджета *Лучшие элементы данных* поддерживаются следующие параметры расширенной настройки.

Parameter	type	name	value
Сортировка узлов сети			
Сортировать по	0	host_ordering_order_by	0 - *(по умолчанию)* Имя узла сети; 3 - Значение элемента данных.
Шаблоны элементов данных	1	host_ordering_item.0	Имя или шаблон элемента данных (например, ”*: Number of processed *values per second”). Примечание: чтобы настроить несколько шаблонов элементов данных, создайте отдельный объект поля виджета панели мониторинга для каждого шаблона элемента данных, увеличивая номер в имени свойства. При настройке виджета на панели мониторинга шаблона следует указывать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*, если для параметра *Сортировать по* установлено значение «Значение элемента данных»

Parameter	type	name	value
Порядок	0	host_ordering_order	2 - *(по умолчанию)* Первые N; 3 - Последние N.
Ограничение	0	host_ordering_limit	Допустимые значения: от 1 до 100. По умолчанию: 10.
Сортировка элементов данных			
Сортировать по	0	item_ordering_order_by	1 - Узел сети; 2 - Имя элемента данных; 3 - *(по умолчанию)* Значение элемента данных.
Шаблоны узлов сети	1	item_ordering_host.0	Имя или шаблон узла сети. Примечание: чтобы настроить несколько шаблонов узлов сети, создайте отдельный объект поля виджета панели мониторинга для каждого шаблона узла сети, увеличивая номер в имени свойства. Поведение параметра: - *обязателен*, если для параметра *Сортировать по* установлено значение «Узел сети»
Порядок	0	item_ordering_order	2 - *(по умолчанию)* Первые N; 3 - Последние N.
Ограничение	0	item_ordering_limit	Допустимые значения: от 1 до 100. По умолчанию: 10.

Столбцы

Note:
Первое число в имени свойства (например, columns.0.items.0, columns.0.item_tags_evaltype) обозначает конкретный столбец, а второе число, если оно указано, обозначает настроенную сущность (например, шаблон элемента данных или тег).

Параметр	тип	имя	значение
Шаблоны элементов данных	1	columns.0.items.0	Имя или шаблон элемента данных (например, ”*: Number of processed *values per second”). Примечание: чтобы настроить несколько шаблонов элементов данных, создайте отдельный объект поля виджета дашборда для каждого шаблона элемента данных, увеличивая второе число в имени свойства. При настройке виджета на дашборде шаблона следует указывать только шаблоны элементов данных, настроенных в шаблоне. Поведение параметра: - *обязательный*
Теги элементов данных			
Тип вычисления	0	columns.0.item_tags_evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.

Параметр	тип	имя	значение
Имя тега	1	columns.0.item_tags.0.tag	Любое строковое значение. Примечание: второе число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательный* при настройке *Тегов элементов данных*
Оператор	0	columns.0.item_tags.0.operator	**0** - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: второе число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательный* при настройке *Тегов элементов данных*
Значение тега	1	columns.0.item_tags.0.value	Любое строковое значение. Примечание: второе число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательный* при настройке *Тегов элементов данных*
Базовый цвет	1	columns.0.base_color	Шестнадцатеричный код цвета (например, `FF0000`).
Отображать значение как	0	columns.0.display_value_as	1 - *(по умолчанию)* Число; 2 - Текст.
Отображение	0	columns.0.display	1 - *(по умолчанию)* Как есть; 2 - Гистограмма; 3 - Индикаторы; 6 - Спарклайн.
Минимум	1	columns.0.min	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”, а для *Отображение* — ”Гистограмма” или ”Индикаторы”
Максимум	1	columns.0.max	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”, а для *Отображение* — ”Гистограмма” или ”Индикаторы”
Спарклайн			
Ширина	0	columns.0.sparkline.width	Допустимые значения находятся в диапазоне от 0 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн”

Параметр	тип	имя	значение
Заполнение	0	columns.0.sparkline.fill	Допустимые значения находятся в диапазоне от 0 до 10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн”
Цвет	1	columns.0.sparkline.color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: `42A5F5`. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн”
Период времени	1	columns.0.sparkline.time_period.reference	`DASHBOARD._timeperiod` - использовать селектор периода времени дашборда в качестве источника данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн”
От	1	columns.0.sparkline.time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн” - *обязательный*, если задан `columns.0.sparkline.time_period.to`
До	1	columns.0.sparkline.time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн” - *обязательный*, если задан `columns.0.sparkline.time_period.from`
Данные истории	0	columns.0.sparkline.history	*(по умолчанию)* Автоматически; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если для *Отображение* установлено значение ”Спарклайн”
Пороговые значения			
Цвет	1	columns.0.thresholds.color	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”

Параметр	тип	имя	значение
Пороговое значение	1	columns.0.thresholds.0.threshold	Пороговое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”
Выделения			
Выделение	1	columns.0.highlights.0.color	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Текст”
Пороговое значение	1	columns.0.highlights.0.pattern	Любое регулярное выражение. Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Текст”
Десятичные разряды	0	columns.0.decimal_places	Допустимые значения находятся в диапазоне от 0 до 10. По умолчанию: 2.
Расширенная настройка			
Функция агрегации	0	columns.0.aggregate_function	Если для *Отображать значение как* установлено значение ”Число”: 0 - *(по умолчанию)* не используется; 1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма; 6 - первое; 7 - последнее. Если для *Отображать значение как* установлено значение ”Текст”: 0 - *(по умолчанию)* не используется; 4 - количество; 6 - первое; 7 - последнее.
Период времени	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - использовать селектор периода времени дашборда в качестве источника данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` Также можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если для *Функция агрегации* установлено значение ”минимум”, ”максимум”, ”среднее”, ”количество”, ”сумма”, ”первое” или ”последнее”

Параметр	тип	имя	значение
От	1	columns.0.time_period.from	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан, а для *Функция агрегации* установлено значение ”минимум”, ”максимум”, ”среднее”, ”количество”, ”сумма”, ”первое” или ”последнее” - *обязательный*, если задан `columns.0.time_period.to`
До	1	columns.0.time_period.to	Допустимая строка времени в абсолютном (`YYYY-MM-DD hh:mm:ss`) или относительном формате (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан, а для *Функция агрегации* установлено значение ”минимум”, ”максимум”, ”среднее”, ”количество”, ”сумма”, ”первое” или ”последнее” - *обязательный*, если задан `columns.0.time_period.from`
Данные истории	0	columns.0.history	0 - *(по умолчанию)* Автоматически; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”
Агрегировать	0	columns.0.aggregate_column	0 - *(по умолчанию)* Каждый элемент данных 1 - Объединенные. Поведение параметра: - *поддерживается*, если для *Отображать значение как* установлено значение ”Число”, а для *Отображение* — ”Как есть”, ”Гистограмма” или ”Индикаторы”.
Функция агрегации объединенных данных	0	columns.0.column_aggregate_function	1 - минимум; 2 - максимум; 3 - среднее; 4 - количество; 5 - сумма *(по умолчанию)*. Поведение параметра: - *обязательный* при настройке *Агрегировать*.
Имя объединенного столбца	1	columns.0.combined_column_name	Любое строковое значение. Поведение параметра: - *обязательный* при настройке *Агрегировать*.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Top items*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Top items*

Настройте виджет *Top items*, который отображает данные для узла сети ”10084” и только для тех элементов данных, у которых тег с именем ”component” содержит значение ”cpu”. Кроме того, отобразите данные так, чтобы узлы сети располагались сверху, и используйте цветную шкалу gauge для представления ячеек.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
```

```
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
                        "name": "Top items",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
```

```
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

30 лучших триггеров

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Топ триггеров* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также повышает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Топ триггеров*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Для виджета *Топ триггеров* поддерживаются следующие параметры.

Параметр	тип	имя	значение
Интервал обновления	0	rf_rate	0 - *(по умолчанию)* Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Группы узлов сети	2	groupids.0	Идентификатор группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	тип	имя	значение
Узлы сети	3	hostids.0	Идентификатор узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* должен быть либо полностью не настроен, либо настроен как минимум с одной группой узлов сети, к которой относятся настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Проблема	1	problem	Имя события проблемы (без учета регистра, полное имя или его часть).
Важность	0	severities.0	Уровни важности триггера. 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. По умолчанию: пусто (все включены). Примечание: чтобы настроить несколько значений, создайте объект поля виджета панели для каждого значения, увеличивая номер в имени свойства.
Теги проблемы			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблемы*
Оператор	0	tags.0.operator	0 - Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует. Примечание: номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблемы*
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: номер в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблемы*

Параметр	тип	имя	значение
Период времени	1	time_period._referenceDASHBOARD._timeperiod	- использовать селектор периода времени панели в качестве источника данных; ABCDE._timeperiod - использовать совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных. По умолчанию: DASHBOARD._timeperiod Также можно задать период времени только в параметрах *От* и *До*.
От	1	time_period.from	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан time_period.to
До	1	time_period.to	Допустимая строка времени в абсолютном (YYYY-MM-DD hh:mm:ss) или относительном формате времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан time_period.from
Ограничение количества триггеров	0	show_lines	Допустимые значения: от 1 до 100. По умолчанию: 10.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Top triggers*. Дополнительную информацию о настройке панели мониторинга см. в dashboard.create.

Настройка виджета *Top triggers*

Настройте виджет *Top triggers*, который отображает 5 триггеров с наибольшим количеством проблем для узла сети "4" с подсчетом всех проблем для каждого триггера. Виджет отображает только триггеры с уровнями серьезности "Warning", "Average", "High" или "Disaster", а также проблемы, у которых есть тег с именем "scope", содержащий значения "performance" или "availability", или "capacity".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
```

```
            "name": "groupids.0",
            "value": 4
        },
        {
            "type": 0,
            "name": "severities.0",
            "value": 2
        },
        {
            "type": 0,
            "name": "severities.1",
            "value": 3
        },
        {
            "type": 0,
            "name": "severities.2",
            "value": 4
        },
        {
            "type": 0,
            "name": "severities.3",
            "value": 5
        },
        {
            "type": 1,
            "name": "tags.0.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.0.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.0.value",
            "value": "performance"
        },
        {
            "type": 1,
            "name": "tags.1.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.1.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.1.value",
            "value": "availability"
        },
        {
            "type": 1,
            "name": "tags.2.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.2.operator",
            "value": 0
```

```
                            },
                            {
                                "type": 1,
                                "name": "tags.2.value",
                                "value": "capacity"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

31 Обзор триггеров

Описание

Эти параметры и возможные значения свойств соответствующих объектов полей виджета позволяют настроить виджет *Обзор триггеров* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Обзор триггеров*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Trigger Overview*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Нет обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
Show	0	show	1 - *(по умолчанию)* Recent problems; 2 - Any; 3 - Problems.
Host groups	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Host groups (Widget)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Hosts	3	hostids.0	ID узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Host groups* должен либо вообще не быть настроен, либо быть настроен как минимум с одной группой узлов сети, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Hosts (Widget/Dashboard)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Problem tags			
Evaluation type	0	evaltype	0 - *(по умолчанию)* And/Or; 2 - Or.
Tag name	1	tags.0.tag	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраивается *Problem tags*

Parameter	type	name	value
Operator	0	tags.0.operator	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраивается *Problem tags*
Tag value	1	tags.0.value	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *обязательно*, если настраивается *Problem tags*
Show suppressed problems	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Layout	0	layout	0 - *(по умолчанию)* Горизонтально; 1 - Вертикально.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Trigger overview*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Обзор триггеров*

Настройте виджет *Обзор триггеров*, который отображает состояния триггеров для всех групп узлов сети, у которых есть триггеры с тегом, имя которого ”scope”, и значением ”availability”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
```

```
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

32 URL

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *URL* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *URL*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *URL*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - *(по умолчанию)* Без обновления; 10 - 10 секунд; 30 - 30 секунд; 60 - 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.
URL	1	url	Допустимая строка URL. Поведение параметра: - *required*
Override host	1	override_hostid._reference	`ABCDE._hostid` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - задайте селектор узлов сети на панели как источник данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели управления для виджета *URL*. Дополнительную информацию о настройке панели управления см. в `dashboard.create`.

Настройка виджета *URL*

Настройте виджет *URL*, который отображает главную страницу руководства Zabbix.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "url",
                        "name": "URL",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "url",
                                "value": "https://www.zabbix.com/documentation/8.0/en"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
```

```
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

33 Веб-мониторинг

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настроить виджет *Веб-мониторинг* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются во время создания или обновления панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск неправильного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Веб-мониторинг*, ознакомьтесь с описанием поведения параметров в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Web monitoring*.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Не обновлять; 10 - 10 секунд; 30 - 30 секунд; 60 - *(по умолчанию)* 1 минута; 120 - 2 минуты; 600 - 10 минут; 900 - 15 минут.

Parameter	type	name	value
Host groups	2	groupids.0	ID группы узлов сети. Примечание: чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Host groups (Widget)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Exclude host groups	2	exclude_groupids.0	ID группы узлов сети. Примечание: чтобы исключить несколько групп узлов сети, создайте объект поля виджета панели для каждой группы узлов сети, увеличивая номер в имени свойства. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Hosts	3	hostids.0	ID узла сети. Примечание: чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети, увеличивая номер в имени свойства. Для нескольких узлов сети параметр *Host groups* должен либо вообще не быть настроен, либо быть настроен как минимум с одной группой узлов сети, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Hosts (Widget/Dashboard)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` - задайте селектор узлов сети панели в качестве источника данных для узлов сети; `ABCDE._hostids` - задайте совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на шаблонной панели.
Scenario tags			
Evaluation type	0	evaltype	0 - *(по умолчанию)* And/Or; 2 - Or.
Tag name	1	tags.0.tag	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Scenario tags*

Parameter	type	name	value
Operator	0	tags.0.operator	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Scenario tags*
Tag value	1	tags.0.value	Любое строковое значение. Примечание: число в имени свойства указывает порядок тега в списке оценки тегов. Поведение параметра: - *required* если настраивается *Scenario tags*
Show hosts in maintenance	0	maintenance	0 - Отключено; 1 - *(по умолчанию)* Включено.
Reference	1	reference	Любое строковое значение, состоящее из 5 символов (например, `ABCDE` или `JBPNL`). Это значение должно быть уникальным в пределах панели, к которой принадлежит виджет. Поведение параметра: - *required*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Web monitoring*. Дополнительную информацию о настройке панели мониторинга см. в `dashboard.create`.

Настройка виджета *Web monitoring*

Настройте виджет *Web monitoring*, который отображает сводку состояния активных сценариев веб-мониторинга для группы узлов сети "4".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
```

```
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Смотрите также

- Поле виджета панели
- `dashboard.create`
- `dashboard.update`

Панель шаблона

Этот класс предназначен для работы с панелями шаблонов.

Ссылки на объекты:

- Панель шаблона
- Страница панели шаблона
    - Виджет панели шаблона
        * Поле виджета панели шаблона

Доступные методы:

- templatedashboard.create - создать новые панели шаблонов
- templatedashboard.delete - удалить панели шаблонов
- templatedashboard.get - получить панели шаблонов
- templatedashboard.update - обновить панели шаблонов

Объект панели шаблона

Следующие объекты напрямую связаны с API `templatedashboard`.

Панель шаблона

Объект панели шаблона имеет следующие свойства.

Property	Type	Description
dashboardid	ID	ID панели шаблона. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя панели шаблона. Поведение свойства: - *обязательно* для операций создания
templateid	ID	ID шаблона, которому принадлежит панель. Поведение свойства: - *константа* - *обязательно* для операций создания
display_period	integer	Интервал слайд-шоу по умолчанию (в секундах). Допустимые значения: 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 30.
auto_start	integer	Автоматически запускать слайд-шоу. Допустимые значения: 0 - не запускать слайд-шоу автоматически; 1 - *(по умолчанию)* запускать слайд-шоу автоматически.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных панелей шаблонов с уже существующими. Автоматически генерируется, если не указан.

Страница панели шаблона

Объект страницы панели шаблона имеет следующие свойства.

Property	Type	Description
dashboard_pageid	ID	ID страницы панели. Поведение свойства: - *только для чтения*
name	string	Имя страницы панели. По умолчанию: пустая строка.
display_period	integer	Интервал слайд-шоу панели (в секундах). Возможные значения: 0, 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 0 (будет использоваться интервал слайд-шоу по умолчанию).
widgets	array	Массив объектов виджета панели шаблона.

Виджет панели шаблона

Объект виджета панели шаблона имеет следующие свойства.

Property	Type	Description
widgetid	ID	ID виджета панели. Property behavior: - *read-only*
type	string	Тип виджета панели. Возможные значения: actionlog - Журнал действий; clock - Часы; discovery - Статус обнаружения; favgraphs - Избранные графики; favmaps - Избранные карты; gauge - Индикатор; graph - График (классический); graphprototype - Прототип графика; honeycomb - Соты; hostavail - Доступность узла сети; hostcard - Карточка узла сети; hostnavigator - Навигатор узлов сети; itemcard - Карточка элемента данных; itemnavigator - Навигатор элементов данных; item - Значение элемента данных; map - Карта; navtree - Дерево навигации по карте; piechart - Круговая диаграмма; plaintext - Обычный текст; problemhosts - Узлы сети с проблемами; problems - Проблемы; problemsbysv - Проблемы по степени тяжести; scatterplot - Точечная диаграмма; slareport - Отчет SLA; svggraph - График; systeminfo - Системная информация; tophosts - Лучшие узлы сети; topitems - Лучшие элементы данных; toptriggers - Лучшие триггеры; trigover - Обзор триггеров; url - URL; web - Мониторинг веб-сайта. Property behavior: - *required*
name	string	Пользовательское имя виджета.
x	integer	Горизонтальная позиция от левой стороны панели. Возможные значения: от 0 до 71.
y	integer	Вертикальная позиция от верхней стороны панели. Возможные значения: от 0 до 63.
width	integer	Ширина виджета. Возможные значения: от 1 до 72.
height	integer	Высота виджета. Возможные значения: от 1 до 64.
view_mode	integer	Режим отображения виджета. Возможные значения: 0 - *(по умолчанию)* стандартный вид виджета; 1 - со скрытым заголовком;
fields	array	Массив объектов поле виджета панели шаблона.

Поле виджета панели шаблона

Объект поля виджета панели шаблона имеет следующие свойства.

Property	Type	Description
type	integer	Тип поля виджета. Возможные значения: 0 - Integer; 1 - String; 4 - Item; 5 - Item prototype; 6 - Graph; 7 - Graph prototype; 8 - Map; 9 - Service; 10 - SLA; 11 - User; 12 - Action; 13 - Media type. Поведение свойства: - *required*
name	string	Имя поля виджета. Возможные значения: см. Поля виджетов панели. Обратите внимание, что некоторые параметры, связанные с узлом сети (например, *Host groups*, *Exclude host groups* и *Hosts* в виджете *Problems*, *Host groups* в виджете *Host availability* и т. д.), недоступны при настройке виджета на панели шаблона. Это связано с тем, что панели шаблона отображают данные только с узла сети, к которому привязан шаблон. Поведение свойства: - *required*
value	mixed	Значение поля виджета в зависимости от типа. Возможные значения: см. Поля виджетов панели. Обратите внимание, что некоторые параметры, связанные с узлом сети (например, *Host groups*, *Exclude host groups* и *Hosts* в виджете *Problems*, *Host groups* в виджете *Host availability* и т. д.), недоступны при настройке виджета на панели шаблона. Это связано с тем, что панели шаблона отображают данные только с узла сети, к которому привязан шаблон. Поведение свойства: - *required*

templatedashboard.create

Описание

`object templatedashboard.create(object/array templateDashboards)`

Этот метод позволяет создавать новые дашборды шаблонов.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Шаблоны панелей мониторинга для создания.

В дополнение к стандартным свойствам шаблона панели мониторинга, метод принимает следующие параметры.

Parameter	Type	Description
pages	array	Страницы шаблона панели мониторинга, которые будут созданы для панели мониторинга. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Поведение параметра: - *required*

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных шаблонов панелей, указанных в свойстве dashboardids. Порядок возвращаемых идентификаторов соответствует порядку переданных шаблонов панелей.

Примеры

Создание панели шаблона

Создайте панель шаблона с именем “Graphs” с одним виджетом Graph на одной странице панели.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.create",
    "params": {
        "templateid": "10318",
        "name": "Graphs",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 6,
                                "name": "graphid",
                                "value": "1123"
                            }
                        ]
                    }
                ]

            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

См. также

- Страница панели шаблона
- Виджет панели шаблона
- Поле виджета панели шаблона

Источник

CTemplateDashboard::create() в *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Описание

`object templatedashboard.delete(array templateDashboardIds)`

Этот метод позволяет удалять шаблоны панелей мониторинга.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых шаблонов панелей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных шаблонов панелей, указанных в свойстве `dashboardids`.

Примеры

Удаление нескольких панелей шаблона

Удаление двух панелей шаблона.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Источник

CTemplateDashboard::delete() в *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Описание

`integer/array templatedashboard.get(object parameters)`

Метод позволяет получать дашборды шаблонов в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object) Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dashboardids	ID/array	Возвращать только шаблонные панели мониторинга с указанными ID.
templateids	ID/array	Возвращать только шаблонные панели мониторинга, принадлежащие указанным шаблонам.
selectPages	query	Возвращать свойство pages со страницами шаблонной панели мониторинга, корректно упорядоченными.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: dashboardid, name.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр countOutput.

Примеры

Получение дашбордов шаблона

Получить все дашборды шаблона с виджетами для указанного шаблона.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "dashboardid": "23",
            "name": "Docker overview",
            "templateid": "10001",
            "display_period": "30",
            "auto_start": "1",
            "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "220",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1125"
                                }
                            ]
                        },
                        {
                            "widgetid": "221",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1129"
                                }
                            ]
                        },
                        {
                            "widgetid": "222",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1128"
                                }
                            ]
```

```
                        },
                        {
                            "widgetid": "223",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1126"
                                }
                            ]
                        },
                        {
                            "widgetid": "224",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "10",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1127"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


См. также

- Страница панели шаблона
- Виджет панели шаблона
- Поле виджета панели шаблона

Источник

CTemplateDashboard::get() в *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Описание

`object templatedashboard.update(object/array templateDashboards)`

Этот метод позволяет обновлять существующие шаблоны панелей.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Свойства шаблона панели мониторинга, которые необходимо обновить.

Для каждой панели мониторинга необходимо указать свойство dashboardid, все остальные свойства необязательны. Будут обновлены только указанные свойства.

Дополнительно к стандартным свойствам шаблона панели мониторинга, метод принимает следующие параметры.

Parameter	Type	Description
pages	array	Страницы шаблона панели мониторинга для замены существующих страниц панели мониторинга. Страницы панели мониторинга обновляются по свойству dashboard_pageid. Новые страницы панели мониторинга будут созданы для объектов без свойства dashboard_pageid, а существующие страницы панели мониторинга будут удалены, если они не используются повторно. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Для свойства pages требуется как минимум один объект страницы панели мониторинга.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы обновленных панелей шаблона в свойстве dashboardids.

Примеры

Переименование панели шаблона

Переименуйте панель шаблона в ”Performance graphs”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Обновление страниц шаблона панели

Переименуйте первую страницу панели, замените виджеты на второй странице панели и добавьте новую страницу в качестве третьей. Удалите все остальные страницы панели.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
```

```
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

См. также

- Виджет панели шаблона
- Поле виджета панели шаблона

Источник

CTemplateDashboard::update() в *ui/include/classes/api/services/CTemplateDashboard.php*.

Пользовательский макрос

Этот класс предназначен для работы с пользовательскими макросами на уровне узла сети и глобальными пользовательскими макросами.

Ссылки на объекты:

- Глобальный макрос
- Макрос узла сети
    - Конфигурация макроса

Доступные методы:

- usermacro.create - создать новые макросы узла сети
- usermacro.createglobal - создать новые глобальные макросы
- usermacro.delete - удалить макросы узла сети
- usermacro.deleteglobal - удалить глобальные макросы
- usermacro.get - получить макросы узла сети и глобальные макросы
- usermacro.update - обновить макросы узла сети
- usermacro.updateglobal - обновить глобальные макросы

Объект пользовательского макроса

Следующие объекты напрямую связаны с API `usermacro`.

Глобальный макрос

Объект глобального макроса имеет следующие свойства.

Property	Type	Description
globalmacroid	ID	ID глобального макроса. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
description	string	Описание макроса.
macro	string	Строка макроса. Поведение свойства: - *обязательно* для операций создания
type	integer	Тип макроса. Возможные значения: 0 - *(по умолчанию)* текстовый макрос; 1 - секретный макрос; 2 - секрет из хранилища.
value	string	Значение макроса. Поведение свойства: - *только для записи* если `type` установлено в "Secret macro" - *обязательно* для операций создания

Макрос узла сети

Объект макроса узла сети определяет макрос, доступный на узле сети, прототипе узла сети или шаблоне. Он имеет следующие свойства.

Property	Type	Description
hostmacroid	ID	ID макроса. Property behavior: - *read-only* - *required* for update operations
automatic	integer	Определяет, управляется ли макрос правилом обнаружения. Possible values: 0 - *(default)* Макрос управляется пользователем; 1 - Макрос управляется правилом обнаружения. Пользователю не разрешено создавать автоматический макрос. Чтобы обновить автоматический макрос, его нужно преобразовать в ручной.
config	object/array	Конфигурация макроса, отвечающая за то, как макрос отображается в Мастере узлов сети.
description	string	Описание макроса.
hostid	ID	ID узла сети, прототипа узла сети или шаблона, которому принадлежит макрос. Property behavior: - *constant* - *required* for create operations

Property	Type	Description
macro	string	Строка макроса. Property behavior: - *required* for create operations
type	integer	Тип макроса. Possible values: 0 - *(default)* Текстовый макрос; 1 - Секретный макрос; 2 - Секрет из хранилища.
value	string	Значение макроса. Property behavior: - *write-only* if `type` is set to ”Secret macro” - *required* for create operations

Настройка макроса

Объект настройки макроса определяет, как макрос отображается в мастере узла сети.

Свойство	Type	Описание
type	integer	Тип поля ввода макроса. Возможные значения: 0 - Макрос не используется в мастере узла сети; 1 - Текстовое поле; 2 - Список; 3 - Флажок. Поведение свойства: - *required*
label	string	Метка для поля ввода макроса. Поведение свойства: - *required*, если `type` задан как ”Textbox”, ”List” или ”Checkbox”
description	text	Справочный текст, отображаемый рядом с полем ввода макроса. Поддерживается форматирование Markdown.
priority	integer	Позиция поля ввода макроса в списке макросов. Если `priority` не задан, макрос добавляется в конец списка несгруппированных макросов.
required	integer	Отмечает макрос как обязательный. Возможные значения: 0 - Необязательный; 1 - Обязательный. Поведение свойства: - *supported*, если `type` задан как ”Textbox” или ”List”
regex	string	Регулярное выражение для проверки пользовательского ввода в текстовом поле. Поведение свойства: - *supported*, если `type` задан как ”Textbox”
section_name	string	Метка сворачиваемого раздела, в котором сгруппирован макрос. Если `section_name` не задан, макрос не группируется. Несгруппированные макросы отображаются первыми; сгруппированные макросы отображаются ниже, упорядоченные по их `priority` в пределах каждого сворачиваемого раздела.

Свойство	Type	Описание
options	text	Строка JSON, определяющая элементы списка или значения флажка. Для списков: массив объектов со свойствами `value` и `text`. Пример: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Для флажков: объект со свойствами `checked` и `unchecked`. Пример: `{"checked": true, "unchecked": false}` Поведение свойства: - *required*, если `type` задан как "List" или "Checkbox"

usermacro.create

Описание

`object usermacro.create(object/array hostMacros)`

Этот метод позволяет создавать новые макросы узла сети.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object/array)` Макросы узла сети для создания.

Метод принимает макросы узла сети со стандартными свойствами макроса узла сети.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных макросов узла сети в свойстве `hostmacroids`. Порядок возвращаемых идентификаторов соответствует порядку переданных макросов узла сети.

Примеры

Создание макроса узла сети

Создайте макрос узла сети `{$SNMP_COMMUNITY}` со значением "public" на узле сети "10198".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::create() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Описание

`object usermacro.createglobal(object/array globalMacros)`

Этот метод позволяет создавать новые глобальные макросы.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object/array)` Глобальные макросы для создания.

Метод принимает глобальные макросы со стандартными свойствами глобальных макросов.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных глобальных макросов в свойстве `globalmacroids`. Порядок возвращаемых идентификаторов соответствует порядку переданных глобальных макросов.

Примеры

Создание глобального макроса

Создайте глобальный макрос `{$SNMP_COMMUNITY}` со значением "public".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::createGlobal() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Описание

`object usermacro.delete(array hostMacroIds)`

Этот метод позволяет удалять макросы узла сети.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых макросов узла сети.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных макросов узла сети, указанных в свойстве `hostmacroids`.

Примеры

Удаление нескольких макросов узла сети

Удаление двух макросов узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::delete() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Описание

`object usermacro.deleteglobal(array globalMacroIds)`

Этот метод позволяет удалять глобальные макросы.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(array)` Идентификаторы глобальных макросов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных глобальных макросов в свойстве `globalmacroids`.

Примеры

Удаление нескольких глобальных макросов

Удаление двух глобальных макросов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::deleteGlobal() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Описание

`integer/array usermacro.get(object parameters)`

Метод позволяет получать макросы узла сети и глобальные макросы в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
globalmacro	flag	Вернуть глобальные макросы вместо макросов узла сети.
globalmacroids	ID/array	Вернуть только глобальные макросы с указанными ID.
groupids	ID/array	Вернуть только макросы узла сети, которые принадлежат узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
hostids	ID/array	Вернуть только макросы, которые принадлежат указанным узлам сети или шаблонам.
hostmacroids	ID/array	Вернуть только макросы узла сети с указанными ID.
inherited	boolean	Если задано `true`, вернуть только пользовательские макросы прототипа узла сети, унаследованные от шаблона.
selectHostGroups	query	Вернуть группы узлов сети, к которым принадлежит макрос узла сети, в свойстве `hostgroups`. Используется только при получении макросов узла сети.

Parameter	Type	Description
selectHosts	query	Вернуть узлы сети, к которым принадлежит макрос узла сети, в свойстве `hosts`.
selectTemplateGroups	query	Используется только при получении макросов узла сети. Вернуть группы шаблонов, к которым принадлежит макрос шаблона, в свойстве `templategroups`.
selectTemplates	query	Используется только при получении макросов шаблона. Вернуть шаблоны, к которым принадлежит макрос узла сети, в свойстве `templates`.
sortfield	string/array	Используется только при получении макросов узла сети. Сортировать результат по указанным свойствам. Возможные значения: `macro`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение макросов узла сети

Получить все макросы узла сети, определенные для узла сети "10198".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "9",
            "hostid": "10198",
            "macro": "{$INTERFACE}",
            "value": "eth0",
            "description": "",
            "type": "0",
```

```
            "automatic": "0"
        },
        {
            "hostmacroid": "11",
            "hostid": "10198",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0",
            "automatic": "0"
        }
    ],
    "id": 1
}
```

Получение макросов узла сети для шаблона

Получить макросы узла сети, определенные для шаблона ”10265”, которые содержат ”STATUS” в имени макроса.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "         IP-                    Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "               Apache",
                "description": "         IP-                    Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "    URL             Apache.",
            "type": "0",
```

```
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "               Apache",
                "description": "   URL            Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "              Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "               Apache",
                "description": "          1   65535        .",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "        ,            HTTP    HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "        ",
                "description": "        ,            HTTP    HTTPS.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Получение глобальных макросов

Получить все глобальные макросы.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

Источник

CUserMacro::get() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Описание

`object usermacro.update(object/array hostMacros)`

Этот метод позволяет обновлять существующие макросы узла сети.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства макроса узла сети, которые необходимо обновить.

Свойство `hostmacroid` должно быть определено для каждого макроса узла сети, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных макросов узла сети в свойстве `hostmacroids`.

Примеры

Изменение значения макроса узла сети

Измените значение макроса узла сети на ”public”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
```

```
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Изменение значения макроса, созданного правилом обнаружения

Преобразуйте созданный правилом обнаружения ”automatic” макрос в ”manual” и измените его значение на ”new-value”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::update() в *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Описание

`object usermacro.updateglobal(object/array globalMacros)`

Этот метод позволяет обновлять существующие глобальные макросы.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства глобального макроса, которые необходимо обновить.

Свойство `globalmacroid` должно быть определено для каждого глобального макроса, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных глобальных макросов в свойстве `globalmacroids`.

Примеры

Изменение значения глобального макроса

Измените значение глобального макроса на "public".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CUserMacro::updateGlobal() в *ui/include/classes/api/services/CUserMacro.php*.

Правило LLD

Этот класс предназначен для работы с правилами обнаружения низкого уровня.

Ссылки на объекты:

- Правило LLD
  - HTTP-заголовок
  - Поле HTTP-запроса
- Фильтр правила LLD
  - Условие фильтра правила LLD
- Путь макроса LLD
- Предобработка правила LLD
- Переопределения правила LLD
  - Фильтр переопределения правила LLD
    - Условие фильтра переопределения правила LLD
  - Операция переопределения правила LLD
    - Статус операции переопределения правила LLD
    - Операция обнаружения переопределения правила LLD
    - Период операции переопределения правила LLD
    - История операции переопределения правила LLD
    - Тренды операции переопределения правила LLD
    - Серьезность операции переопределения правила LLD
    - Тег операции переопределения правила LLD
    - Шаблон операции переопределения правила LLD
    - Инвентаризация операции переопределения правила LLD

Доступные методы:

- discoveryrule.create - создать новые правила LLD
- discoveryrule.delete - удалить правила LLD
- discoveryrule.get - получить правила LLD
- discoveryrule.update - обновить правила LLD

Объект правила LLD

Следующие объекты напрямую связаны с API `discoveryrule`.

Правило LLD

Объект правила низкоуровневого обнаружения имеет следующие свойства.

Свойство	Тип	Описание
itemid	ID	ID правила LLD. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
delay	string	Интервал обновления правила LLD. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы. Если используется макрос, значение должно состоять из одного макроса. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы можно записывать как два макроса, разделённых косой чертой (например, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Пример: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Поведение свойства: - *обязательно*, если `type` имеет значение ”Zabbix agent” (0), ”Simple check” (3), ”Zabbix internal” (5), ”External check” (10), ”Database monitor” (11), ”IPMI agent” (12), ”SSH agent” (13), ”TELNET agent” (14), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”Browser” (22) или если `type` имеет значение ”Zabbix agent (active)” (7), а `key_` не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит правило LLD. Поведение свойства: - *постоянное* - *обязательно* для операций создания
flags	integer	Источник правила обнаружения. Возможные значения: 1 - правило низкоуровневого обнаружения; 5 - правило низкоуровневого обнаружения, преобразованное из прототипа. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
interfaceid	ID	ID интерфейса узла сети правила LLD. Поведение свойства: - *обязательно*, если правило LLD принадлежит узлу сети, а `type` имеет значение ”Zabbix agent”, ”IPMI agent”, ”JMX agent” или ”SNMP agent” - *поддерживается*, если правило LLD принадлежит узлу сети, а `type` имеет значение ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”
key_	string	Ключ правила LLD. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
name	string	Имя правила LLD. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
type	integer	Тип правила LLD. Возможные значения: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Этот тип разрешён только в том случае, если родителем правила является шаблон (предполагается, что он связан с прототипом узла сети при обнаружении) или обнаруженный узел сети (у которого корневым объектом является правило LLD). Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
url	string	Строка URL. Поддерживает пользовательские макросы, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
allow_traps	integer	Разрешить заполнение значения аналогично элементу данных типа trapper. Возможные значения: 0 - *(по умолчанию)* не принимать входящие данные; 1 - принимать входящие данные. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent”
authtype	integer	Метод аутентификации. Возможные значения, если `type` имеет значение ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если `type` имеет значение ”HTTP agent”: 0 - *(по умолчанию)* отсутствует; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если `type` имеет значение ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)
description	string	Описание правила LLD.
error	string	Текст ошибки, если при обновлении значения правила LLD возникли проблемы. Поведение свойства: - *только для чтения*
follow_redirects	integer	Следовать перенаправлениям ответа при опросе данных. Возможные значения: 0 - не следовать перенаправлениям; 1 - *(по умолчанию)* следовать перенаправлениям. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
headers	array	Массив заголовков, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
http_proxy	string	Строка подключения HTTP(S)-прокси. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Датчик IPMI. Поведение свойства: - *обязательно*, если `type` имеет значение ”IPMI agent”, а `key_` не имеет значения ”ipmi.get” - *поддерживается*, если `type` имеет значение ”IPMI agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
jmx_endpoint	string	Пользовательская строка подключения агента JMX. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` имеет значение ”JMX agent”
lifetime	string	Период времени, по истечении которого элементы данных, больше не обнаруживаемые, будут удалены. Принимает количество секунд (например, 3600), значение с суффиксом времени (например, 3600s, 180m, 24h, 1d) или пользовательский макрос. По умолчанию: 7d.
lifetime_type	integer	Сценарий удаления потерянных ресурсов LLD. Возможные значения: 0 - *(по умолчанию)* удалить после достижения порога времени хранения; 1 - не удалять; 2 - удалить немедленно.
enabled_lifetime	string	Период времени, по истечении которого элементы данных, больше не обнаруживаемые, будут отключены. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательский макрос. По умолчанию: 0.
enabled_lifetime_type	integer	Сценарий отключения потерянных ресурсов LLD. Возможные значения: 0 - отключить после достижения порога времени хранения; 1 - не отключать; 2 - *(по умолчанию)* отключить немедленно.
master_itemid	ID	ID главного элемента данных. Правило обнаружения не может быть главным элементом данных для другого правила обнаружения. Поведение свойства: - *обязательно*, если `type` имеет значение ”Dependent item” - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* сохранить без изменений; 1 - преобразовать в JSON.
params	string	Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов Дополнительные параметры в зависимости от типа правила LLD: - выполняемый скрипт для правил LLD SSH и Telnet; - SQL-запрос для правил LLD мониторинга баз данных; - формула для вычисляемых правил LLD; - скрипт для правил LLD типа Script и Browser. Поведение свойства: - *обязательно*, если `type` имеет значение ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”Script” или ”Browser”)

Свойство	Тип	Описание
parameters	object/array	Дополнительные параметры, если `type` имеет значение ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должен быть уникальным. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Script” или ”Browser” - *только для чтения* для унаследованных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”JMX agent”, а `username` задан - *поддерживается*, если `type` имеет значение ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)
post_type	integer	Тип тела данных POST, хранящегося в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* необработанные данные; 2 - данные JSON; 3 - данные XML. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”JSON data” или ”XML data” - *поддерживается*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”Raw data” - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
query_fields	array	Массив полей запроса, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
request_method	integer	Тип метода запроса. Возможные значения: 0 - *(по умолчанию)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
retrieve_mode	integer	Какая часть ответа должна быть сохранена. Возможные значения, если `request_method` имеет значение ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* тело; 1 - заголовки; 2 - сохранить тело и заголовки. Возможные значения, если `request_method` имеет значение ”HEAD”: 1 - заголовки. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
snmp_oid	string	OID SNMP. Поведение свойства: - *обязательно*, если `type` имеет значение ”SNMP agent” - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль файла ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
state	integer	Состояние правила LLD. Возможные значения: 0 - *(по умолчанию)* нормальное; 1 - не поддерживается. Поведение свойства: - *только для чтения*
status	integer	Состояние правила LLD. Возможные значения: 0 - *(по умолчанию)* включённое правило LLD; 1 - отключённое правило LLD.

Свойство	Тип	Описание
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Также поддерживает пользовательские макросы как часть списка, разделённого запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
templateid	ID	ID родительского правила LLD шаблона. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса на опрос данных элемента данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-600s. По умолчанию: "" - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` имеет значение "Zabbix agent" (0), "Simple check" (3), а `key_` не начинается с "`vmware.`" и "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20), а `snmp_oid` начинается с "`walk[`" или "`get[`", "Script" (21), "Browser" (22) - *только для чтения* для унаследованных объектов, - если применяется значение по умолчанию, а элемент данных принадлежит узлу сети, который отслеживается недоступным прокси или группой прокси, используется глобальный тайм-аут.
trapper_hosts	string	Разрешённые узлы сети. Поведение свойства: - *поддерживается*, если `type` имеет значение "Zabbix trapper" или если `type` имеет значение "HTTP agent", а `allow_traps` имеет значение "Allow incoming data". Если свойство не указано, ему автоматически присваивается пользовательский макрос `{$TRAPPER.ALLOWED_HOSTS}` (значение по умолчанию). По умолчанию `{$TRAPPER.ALLOWED_HOSTS}` имеет значение `127.0.0.1,::1`. Если свойству присвоена пустая строка, входящие подключения от всех узлов сети будут запрещены.
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение "SSH agent", "TELNET agent" или если `type` имеет значение "JMX agent", а `password` задан - *поддерживается*, если `type` имеет значение "Simple check", "Database monitor" или "HTTP agent" - *только для чтения* для унаследованных объектов (если `type` имеет значение "HTTP agent")
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных правил LLD с уже существующими. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если правило LLD принадлежит шаблону

Свойство	Тип	Описание
verify_host	integer	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
verify_peer	integer	Проверять ли подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

HTTP-заголовок

Объект заголовка имеет следующие свойства:

Property	Type	Description
name	string	Имя HTTP-заголовка. Поведение свойства: - *required*
value	string	Значение заголовка. Поведение свойства: - *required*

Поле HTTP-запроса

Объект поля запроса определяет имя и значение, которые используются для указания параметра URL. Он имеет следующие свойства:

Property	Type	Description
name	string	Имя параметра. Property behavior: - *required*
value	string	Значение параметра. Property behavior: - *required*

Фильтр правила LLD

Объект фильтра правила LLD определяет набор условий, которые можно использовать для фильтрации обнаруженных объектов. Он имеет следующие свойства:

Property	Type	Description
conditions	object/array	Набор условий фильтра, используемых для фильтрации результатов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *required*

Property	Type	Description
evaltype	integer	Метод оценки условий фильтра. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Пользовательское выражение. Поведение свойства: - *required*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для оценки условий фильтра. Выражение содержит идентификаторы, которые ссылаются на конкретные условия фильтра через их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *read-only*
formula	string	Определяемое пользователем выражение, используемое для оценки условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются на конкретные условия фильтра через их `formulaid`. Идентификаторы, используемые в выражении, должны в точности совпадать с теми, что определены в условиях фильтра: ни одно условие не может остаться неиспользованным или быть пропущенным. Поведение свойства: - *required* если `evaltype` установлен в ”custom expression”

Условие фильтра правила LLD

Объект условия фильтра правила LLD определяет отдельную проверку, выполняемую для значения макроса LLD. Он имеет следующие свойства:

Property	Type	Description
macro	string	Макрос LLD, для которого выполняется проверка. Property behavior: - *required*
value	string	Значение для сравнения. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть задан пользователем при изменении условий фильтра, но будет сгенерирован заново при последующем запросе. Property behavior: - *required* if `evaltype` of LLD rule filter object is set to ”custom expression”

Property	Type	Description
operator	integer	Оператор условия. Possible values: 8 - *(default)* matches regular expression; 9 - does not match regular expression; 12 - exists; 13 - does not exist.

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, см. примеры на страницах методов discoveryrule.get и discoveryrule.create.

Путь макроса LLD

Путь макроса LLD имеет следующие свойства:

Property	Type	Description
lld_macro	string	Макрос LLD. Поведение свойства: - *required*
path	string	Селектор для значения, которое будет назначено соответствующему макросу. Поведение свойства: - *required*

Предобработка правила LLD

Объект предобработки правила LLD имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип параметра предобработки. Возможные значения: 5 - Регулярное выражение; 11 - XML XPath; 12 - JSONPath; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка наличия ошибки в JSON; 17 - Проверка наличия ошибки в XML; 20 - Отбрасывать неизменившиеся значения с контролем доступности; 21 - JavaScript; 23 - Prometheus в JSON; 24 - CSV в JSON; 25 - Замена; 27 - XML в JSON; 28 - Значение SNMP walk; 29 - SNMP walk в JSON; 30 - Получение значения SNMP. Поведение свойства: - *обязательное*

Свойство	Тип	Описание
params	string	Дополнительные параметры, используемые параметром предобработки. Несколько параметров разделяются символом новой строки (\n). Поведение свойства: - *обязательное*, если для `type` установлено значение «Регулярное выражение» (5), «XML XPath» (11), «JSONPath» (12), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Отбрасывать неизменившиеся значения с контролем доступности» (20), «JavaScript» (21), «Prometheus в JSON» (23), «CSV в JSON» (24), «Замена» (25), «Значение SNMP walk» (28), «SNMP walk в JSON» (29) или «Получение значения SNMP» (30)
error_handler	integer	Тип действия, используемый в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке задается сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательное*, если для `type` установлено значение «Регулярное выражение» (5), «XML XPath» (11), «JSONPath» (12), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Prometheus в JSON» (23), «CSV в JSON» (24), «XML в JSON» (27), «Значение SNMP walk» (28), «SNMP walk в JSON» (29) или «Получение значения SNMP» (30)
error_handler_params	string	Параметры обработчика ошибок. Поведение свойства: - *обязательное*, если для `error_handler` установлено значение «Установить пользовательское значение» или «Установить пользовательское сообщение об ошибке»

Для каждого типа предобработки поддерживаются следующие параметры и обработчики ошибок.

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
5	Регулярное выражение	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Соответствует регулярному выражению	pattern[1]			0, 1, 2, 3
15	Не соответствует регулярному выражению	pattern[1]			0, 1, 2, 3
16	Проверка наличия ошибки в JSON	path[3]			0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
17	Проверка наличия ошибки в XML	path[3]			0, 1, 2, 3
20	Отбрасывать неизменившиеся значения с контролем доступности	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus в JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV в JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Замена	search string[2]	replacement[2]		
27	XML в JSON				0, 1, 2, 3
28	Значение SNMP walk	OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS		0, 1, 2, 3
29	SNMP walk в JSON[7]	Field name[2]	OID prefix[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS	0, 1, 2, 3
30	Получение значения SNMP	Формат: 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS			0, 1, 2, 3

[1] регулярное выражение
 [2] строка
 [3] JSONPath или XML XPath
 [4] положительное целое число (с поддержкой суффиксов времени, например 30s, 1m, 2h, 1d)
 [5] пользовательский макрос
 [6] шаблон Prometheus, соответствующий синтаксису: `<metric name>{<label name>="<label value>", ...} == <value>`. Каждый компонент шаблона Prometheus (метрика, имя метки, значение метки и значение метрики) может быть пользовательским макросом.
 [7] Поддерживает несколько записей «Имя поля, префикс OID, записи формата», разделенных символом новой строки.

Переопределения правил LLD

Объект переопределений правил LLD определяет набор правил (фильтры, условия и операции), которые используются для переопределения свойств различных объектов прототипов. Он имеет следующие свойства:

Property	Type	Description
name	string	Уникальное имя переопределения. Property behavior: - *required*
step	integer	Уникальный порядковый номер переопределения. Property behavior: - *required*
stop	integer	Остановить обработку следующих переопределений, если совпадает. Possible values: 0 - *(default)* не останавливать обработку переопределений; 1 - остановить обработку переопределений, если фильтр совпадает.
filter	object	Фильтр переопределения.
operations	object/array	Операции переопределения.

Фильтр переопределения правила LLD

Объект фильтра переопределения правила LLD определяет набор условий, и если они совпадают с обнаруженным объектом, применяется переопределение. Он имеет следующие свойства:

Property	Type	Description
conditions	object/array	Набор условий фильтра переопределения, используемых для сопоставления обнаруженных объектов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *required*
evaltype	integer	Метод оценки условий фильтра переопределения. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Поведение свойства: - *required*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для оценки условий фильтра переопределения. Выражение содержит идентификаторы, которые ссылаются на конкретные условия фильтра переопределения по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *read-only*

Property	Type	Description
formula	string	Определяемое пользователем выражение, используемое для оценки условий фильтров переопределения с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются на конкретные условия фильтра переопределения по их `formulaid`. Идентификаторы, используемые в выражении, должны в точности совпадать с теми, что определены в условиях фильтра переопределения: ни одно условие не может остаться неиспользованным или быть пропущенным. Поведение свойства: - *required* if `evaltype` is set to "custom expression"

Условие фильтра переопределения правила LLD

Объект условия фильтра переопределения правила LLD определяет отдельную проверку, выполняемую для значения макроса LLD. Он имеет следующие свойства:

Property	Type	Description
macro	string	Макрос LLD, для которого выполняется проверка. Property behavior: - *required*
value	string	Значение для сравнения. Property behavior: - *required* if `operator` is set to "matches regular expression" or "does not match regular expression"
formulaid	string	Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть задан пользователем при изменении условий фильтра, но будет сгенерирован заново при последующем запросе. Property behavior: - *required* if `evaltype` of LLD rule override filter object is set to "custom expression"
operator	integer	Оператор условия. Possible values: 8 - *(default)* matches regular expression; 9 - does not match regular expression; 12 - exists; 13 - does not exist.

Операция переопределения правила LLD

Операция переопределения правила LLD представляет собой комбинацию условий и действий, выполняемых над объектом-прототипом. Она имеет следующие свойства:

Property	Type	Description
operationobject	integer	Тип обнаруженного объекта, для которого выполняется действие. Возможные значения: 0 - прототип элемента данных; 1 - прототип триггера; 2 - прототип графика; 3 - прототип узла сети; 4 - прототип обнаружения. Поведение свойства: - *required*
operator	integer	Оператор условия переопределения. Возможные значения: 0 - *(по умолчанию)* равно; 1 - не равно; 2 - содержит; 3 - не содержит; 8 - соответствует; 9 - не соответствует.
value	string	Шаблон для сопоставления с именем прототипа элемента данных, триггера, графика или узла сети в зависимости от выбранного объекта.
opstatus	object	Объект операции переопределения статуса для объектов-прототипов элемента данных, триггера и узла сети.
opdiscover	object	Объект операции переопределения статуса обнаружения (для всех типов объектов).
opperiod	object	Объект операции переопределения периода (интервала обновления) для объекта-прототипа элемента данных.
ophistory	object	Объект операции переопределения истории для объекта-прототипа элемента данных.
optrends	object	Объект операции переопределения трендов для объекта-прототипа элемента данных.
opseverity	object	Объект операции переопределения серьезности для объекта-прототипа триггера.
optag	object/array	Объект операции переопределения тега для объектов-прототипов триггера и узла сети.
optemplate	object/array	Объект операции переопределения шаблона для объекта-прототипа узла сети.
opinventory	object	Объект операции переопределения инвентаря для объекта-прототипа узла сети.

Статус операции переопределения правила LLD

Статус операции переопределения правила LLD, который задается для обнаруженного объекта. Он имеет следующие свойства:

Property	Type	Description
status	integer	Переопределяет статус для выбранного объекта. Возможные значения: 0 - Создать включенным; 1 - Создать отключенным. Поведение свойства: *- required*

Операция переопределения правила LLD: discover

Статус операции переопределения правила LLD discover, который задается для обнаруженного объекта. Он имеет следующие свойства:

Property	Type	Description
discover	integer	Переопределяет статус discover для выбранного объекта. Возможные значения: 0 - Да, продолжать обнаружение объектов; 1 - Нет, новые объекты не будут обнаруживаться, а существующие будут помечены как потерянные. Поведение свойства: *- required*

Период выполнения операции переопределения правила LLD

Период выполнения операции переопределения правила LLD — это значение интервала обновления, заданное для обнаруженного элемента данных. Он имеет следующие свойства:

Property	Type	Description
delay	string	Переопределяет интервал обновления прототипа элемента данных. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы или макросы LLD. При использовании значение должно состоять из одного макроса. Несколько макросов или макросы в сочетании с текстом не поддерживаются. Гибкие интервалы можно записывать в виде двух макросов, разделённых косой чертой (например, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Пример: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Поведение свойства: *- обязательное*

История операций переопределения правила LLD

Значение истории операций переопределения правила LLD, которое задается для обнаруженного элемента данных. Оно имеет следующие свойства:

Property	Type	Description
history	string	Переопределяет историю прототипа элемента данных, которая представляет собой единицу времени, определяющую, как долго должны храниться данные истории. Также принимает пользовательский макрос и LLD-макрос. Property behavior: - *required*

Тренды операции переопределения правила LLD

Значение трендов операции переопределения правила LLD, которое задается для обнаруженного элемента данных. Оно имеет следующие свойства:

Property	Type	Description
trends	string	Переопределяет тренды прототипа элемента данных, то есть задает единицу времени, в течение которой должны храниться данные трендов. Также принимает пользовательский макрос и макрос LLD. Property behavior: - *required*

Переопределение серьезности операции правила LLD

Значение серьезности операции переопределения правила LLD, которое задается для обнаруженного триггера. Оно имеет следующие свойства:

Property	Type	Description
severity	integer	Переопределяет серьезность прототипа триггера. Возможные значения: 0 - *(по умолчанию)* не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа. Поведение свойства: - *required*

Тег операции переопределения правила LLD

Объект тега операции переопределения правила LLD содержит имя тега и значение, которые задаются для обнаруженного объекта. Он имеет следующие свойства:

Property	Type	Description
tag	string	Новое имя тега. Также принимает пользовательский макрос и LLD-макрос. Property behavior: - *required*
value	string	Новое значение тега. Также принимает пользовательский макрос и LLD-макрос.

Шаблон операции переопределения правила LLD

Объект шаблона операции переопределения правила LLD, который связан с обнаруженным узлом сети. Он имеет следующие свойства:

Property	Type	Description
templateid	ID	Переопределяет шаблон связанных шаблонов прототипа узла сети. Property behavior: - *required*

Инвентарь операции переопределения правила LLD

Значение режима инвентаря операции переопределения правила LLD, которое задается для обнаруженного узла сети. Имеет следующие свойства:

Property	Type	Description
inventory_mode	integer	Переопределяет режим инвентаря прототипа узла сети. Возможные значения: -1 - отключено; 0 - *(по умолчанию)* вручную; 1 - автоматически. Поведение свойства: - *required*

discoveryrule.create

Описание

`object discoveryrule.create(object/array lldRules)`

Этот метод позволяет создавать новые правила LLD.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Правила LLD для создания.

В дополнение к стандартным свойствам правила LLD, метод принимает следующие параметры.

Parameter	Type	Description
filter	object	Фильтр правила LLD для правила LLD.
preprocessing	object/array	Параметры предобработки правила LLD.
lld_macro_paths	object/array	Параметры lld_macro_path правила LLD.
overrides	object/array	Параметры переопределений правила LLD.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных правил LLD, указанных в свойстве `itemids`. Порядок возвращаемых ID совпадает с порядком переданных правил LLD.

Примеры

Создание правила LLD

Создайте правило LLD агента Zabbix для обнаружения смонтированных файловых систем. Обнаруженные элементы данных будут обновляться каждые 30 секунд.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Mounted filesystem discovery",
        "key_": "vfs.fs.discovery",
        "hostid": "10197",
        "type": 0,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Использование фильтра

Создайте правило LLD с набором условий для фильтрации результатов. Условия будут объединены с помощью логического оператора "and".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
```

```
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Создание правила LLD с путями макросов

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "    LLD            LLD",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
            {
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Использование фильтра с пользовательским выражением

Создайте правило LLD с фильтром, который будет использовать пользовательское выражение для оценки условий. Правило LLD должно обнаруживать только те объекты, у которых значение макроса "{#MACRO1}" соответствует одновременно обоим регулярным выражениям "regex1" и "regex2", а значение "{#MACRO2}" соответствует либо "regex3", либо "regex4". Идентификаторы формул "A", "B", "C" и "D" выбраны произвольно.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Использование пользовательских полей запроса и заголовков

Создайте правило LLD с пользовательскими полями запроса и заголовками.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
```

```
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
            {
                "name": "X-Type",
                "value": "api"
            },
            {
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Создание правила LLD с предварительной обработкой

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discovery rule with preprocessing",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Создание правила LLD с переопределениями

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discover database host",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Discover MySQL host",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mysqld\\.service$"
                        },
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mariadb\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10170"
                            }
                        ],
                        "optag": [
                            {
```

```
                                "tag": "database",
                                "value": "mysql"
                            }
                        ]
                    }
                ]
            },
            {
                "name": "Discover PostgreSQL host",
                "step": "2",
                "stop": "1",
                "filter": {
                    "evaltype": "0",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^postgresql\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Создание правила LLD для скрипта

Создайте простой сбор данных с помощью правила LLD для скрипта.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.lldrule",
        "hostid": "12345",
        "type": 21,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [{
            "name": "host",
            "value": "{HOST.CONN}"
        }],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Создание правила LLD с указанным периодом времени для отключения и без удаления

Создайте правило LLD с настраиваемым периодом времени для отключения сущности после того, как она больше не обнаруживается, с настройкой, при которой она никогда не будет удалена.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "lld disable after 1h",
        "key_": "lld.disable",
        "hostid": "10001",
        "type": 2,
        "lifetime_type": 1,
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "1h"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

См. также

- Фильтр правила LLD
- Пути макросов LLD
- Предобработка правила LLD

Источник

CDiscoveryRule::create() в *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Описание

`object discoveryrule.delete(array lldRuleIds)`

Этот метод позволяет удалить правила обнаружения низкого уровня (LLD).

> **Note:**
> Этот метод доступен только пользователям с типом *Admin* и *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.
> Пользователь не сможет удалить правило LLD, которое обнаруживает сущности с использованием недоступного прокси. Дополнительную информацию см. в разделе Настройка групп пользователей (*Список доступа к прокси*).

Параметры

( ) ID удаляемых правил LLD.

Возвращаемые значения

( ) Возвращает объект, который содержит идентификаторы удаленных правил LLD, указанных в свойстве `ruleids`.

Примеры

Удаление нескольких правил LLD

Удалить два правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
    "27665",
    "27668"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
     "ruleids": [
     "27665",
     "27668"
    ]
    },
    "id": 1
}
```

Источник

CDiscoveryRule::delete() в *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Описание

`integer/array discoveryrule.get(object parameters)`

Метод позволяет получать правила LLD в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
itemids	ID/array	Возвращать только правила LLD с указанными ID.
groupids	ID/array	Возвращать только правила LLD, которые принадлежат узлам сети из указанных групп.
hostids	ID/array	Возвращать только правила LLD, которые принадлежат указанным узлам сети.
inherited	boolean	Если установлено значение `true`, возвращать только правила LLD, унаследованные от шаблона.
interfaceids	ID/array	Возвращать только правила LLD, использующие указанные интерфейсы узлов сети.
monitored	boolean	Если установлено значение `true`, возвращать только включенные правила LLD, которые принадлежат контролируемым узлам сети.
templated	boolean	Если установлено значение `true`, возвращать только правила LLD, которые принадлежат шаблонам.
templateids	ID/array	Возвращать только правила LLD, которые принадлежат указанным шаблонам.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения правила LLD. Объект обнаружения правила LLD связывает обнаруженное правило LLD с прототипом правила LLD, из которого оно было обнаружено.
selectDiscoveryRulePrototypes	query	Возвращать свойство `discoveryRulePrototypes` с прототипами правил LLD, которые принадлежат правилу LLD. Поддерживает `count`.
selectFilter	query	Возвращать свойство `filter` с данными фильтра, используемого правилом LLD.
selectGraphs	query	Возвращает свойство `graphs` с прототипами графиков, которые принадлежат правилу LLD. Поддерживает `count`.
selectHostPrototypes	query	Возвращать свойство `hostPrototypes` с прототипами узлов сети, которые принадлежат правилу LLD. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, к которым принадлежит правило LLD.
selectItems	query	Возвращать свойство `items` с прототипами элементов данных, которые принадлежат правилу LLD. Поддерживает `count`.
selectTriggers	query	Возвращать свойство `triggers` с прототипами триггеров, которые принадлежат правилу LLD. Поддерживает `count`.
selectLLDMacroPaths	query	Возвращать свойство `lld_macro_paths` со списком макросов LLD и путей к значениям, назначенным каждому соответствующему макросу.
selectPreprocessing	query	Возвращать свойство `preprocessing` с параметрами предварительной обработки правила LLD.

Parameter	Type	Description
selectOverrides	query	Возвращать свойство `lld_rule_overrides` со списком фильтров переопределения, условий и операций, выполняемых над объектами прототипов.
filter	object	Возвращать только те результаты, которые точно соответствуют заданному фильтру. Принимает объект, где ключи - это имена свойств, а значения - либо одно значение, либо массив значений, с которыми выполняется сопоставление. Не поддерживает свойства с типом данных `text` data type. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит правило LLD.
limitSelects	integer	Ограничивает количество записей, возвращаемых подвыборками. Применяется к следующим подвыборкам: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение правил обнаружения с узла сети

Получить все правила обнаружения для указанного ID узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "                    ",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "                    ,                                    \"Network interfaces for disc
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        },
        {
            "itemid": "27426",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "                          ",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
```

```
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "                                ,                                \"File systems for dis
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


Получение условий фильтра

Получите имя правила LLD "24681" и его условия фильтра.

Фильтр использует тип оценки "and", поэтому свойство `formula` пустое, а `eval_formula` генерируется автоматически.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "itemid": "24681",
            "name": "Filtered LLD rule",
            "filter": {
                "evaltype": "1",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#MACRO1}",
                        "value": "@regex1",
                        "operator": "8",
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#MACRO2}",
                        "value": "@regex2",
                        "operator": "9",
                        "formulaid": "B"
                    },
                    {
                        "macro": "{#MACRO3}",
                        "value": "",
                        "operator": "12",
                        "formulaid": "C"
                    },
                    {
                        "macro": "{#MACRO4}",
                        "value": "",
                        "operator": "13",
                        "formulaid": "D"
                    }
                ],
                "eval_formula": "A and B and C and D"
            }
        }
    ],
    "id": 1
}
```

Получение правила LLD по URL

Получение правила LLD для узла сети по значению поля URL правила. Поддерживается только точное совпадение строки URL, заданной для правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "itemid": "28336",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10257",
            "name": "API HTTP agent",
            "key_": "api_discovery_rule",
            "delay": "5s",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
                    "value": "api"
                },
                {
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
```

```
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```

Получение правила LLD с переопределениями

Получение одного правила LLD, для которого заданы различные параметры переопределения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
                                "formulaid": "A"
                            },
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mariadb\\.service$",
                                "formulaid": "B"
                            }
                        ],
                        "eval_formula": "A or B"
                    },
                    "operations": [
                        {
                            "operationobject": "3",
                            "operator": "2",
                            "value": "Database host",
                            "opstatus": {
```

```
                                "status": "0"
                            },
                            "optag": [
                                {
                                    "tag": "database",
                                    "value": "mysql"
                                }
                            ],
                            "optemplate": [
                                {
                                    "templateid": "10170"
                                }
                            ]
                        }
                    ]
                },
                {
                    "name": "Discover PostgreSQL host",
                    "step": "2",
                    "stop": "1",
                    "filter": {
                        "evaltype": "0",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^postgresql\\.service$",
                                "formulaid": "A"
                            }
                        ],
                        "eval_formula": "A"
                    },
                    "operations": [
                        {
                            "operationobject": "3",
                            "operator": "2",
                            "value": "Database host",
                            "opstatus": {
                                "status": "0"
                            },
                            "optag": [
                                {
                                    "tag": "database",
                                    "value": "postgresql"
                                }
                            ],
                            "optemplate": [
                                {
                                    "templateid": "10263"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


См. также

- Прототип графика
- Узел сети
- Прототип элемента данных
- Фильтр правила LLD
- Прототип триггера

Источник

CDiscoveryRule::get() в *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Описание

`object discoveryrule.update(object/array lldRules)`

Этот метод позволяет обновлять существующие правила LLD.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства правила LLD, которые необходимо обновить.

Для каждого правила LLD должно быть определено свойство `itemid`, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств правила LLD, метод принимает следующие параметры.

Parameter	Type	Description
filter	object	Фильтр правила LLD для замены существующего фильтра.
preprocessing	object/array	Параметры предобработки правила LLD для замены существующих параметров предобработки. Поведение параметра: - *только для чтения* для унаследованных объектов
lld_macro_paths	object/array	Параметры lld_macro_path правила LLD для замены существующих параметров lld_macro_path. Поведение параметра: - *только для чтения* для унаследованных объектов
overrides	object/array	Параметры переопределений правила LLD для замены существующих параметров переопределений. Поведение параметра: - *только для чтения* для унаследованных объектов

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных правил LLD, указанных в свойстве `itemids`.

Примеры

Добавление фильтра к правилу LLD

Добавьте фильтр, чтобы содержимое макроса *{#FSTYPE}* соответствовало регулярному выражению *@File systems for discovery*.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
```

```
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Добавление путей макросов LLD

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Отключение trapping

Отключить LLD trapping для правила обнаружения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
```

```
        "allow_traps": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Обновление параметров предварительной обработки правила LLD

Обновите правило LLD с правилом предварительной обработки "JSONPath".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.path.to.json",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Обновление скрипта правила LLD

Обновите скрипт правила LLD, указав другой скрипт, и удалите ненужные параметры, которые использовались предыдущим скриптом.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Обновление срока жизни правила LLD

Обновите правило LLD, чтобы отключать больше не обнаруживаемый объект через 12 часов и удалять его через 7 дней.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Источник

CDiscoveryRule::update() в *ui/include/classes/api/services/CDiscoveryRule.php*.

Правило обнаружения

Этот класс предназначен для работы с правилами обнаружения сети.

> **Note:**
> Этот API предназначен для работы с правилами обнаружения сети. Для правил низкоуровневого обнаружения см. API правила LLD.

Ссылки на объекты:

- Правило обнаружения

Доступные методы:

- drule.create - создать новые правила обнаружения сети
- drule.delete - удалить правила обнаружения сети
- drule.get - получить правила обнаружения сети

- drule.update - обновить правила обнаружения сети

Объект правила обнаружения

Следующие объекты напрямую связаны с API `drule`.

Правило обнаружения

Объект правила обнаружения определяет правило обнаружения сети. Он имеет следующие свойства.

Property	Type	Description
druleid	ID	Идентификатор правила обнаружения. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
iprange	string	Один или несколько диапазонов IP-адресов для проверки, разделенных запятыми. Дополнительную информацию о поддерживаемых форматах диапазонов IP-адресов см. в разделе настройка обнаружения сети. Поведение свойства: - *обязательно* для операций создания
name	string	Имя правила обнаружения. Поведение свойства: - *обязательно* для операций создания
delay	string	Интервал выполнения правила обнаружения. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или пользовательский макрос. По умолчанию: 1h.
proxyid	ID	Идентификатор прокси, используемого для обнаружения.
status	integer	Определяет, включено ли правило обнаружения. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.
concurrency_max	integer	Максимальное количество параллельных проверок для каждого правила обнаружения. Возможные значения: 0 - *(по умолчанию)* неограниченное количество проверок; 1 - одна проверка; 2-999 - пользовательское количество проверок.
error	string	Текст ошибки, если при выполнении правила обнаружения возникли проблемы. Поведение свойства: - *только для чтения*

drule.create

Описание

`object drule.create(object/array discoveryRules)`

Этот метод позволяет создавать новые правила обнаружения.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Правила обнаружения для создания.

В дополнение к стандартным свойствам правила обнаружения, метод принимает следующие параметры.

Параметр	Тип	Описание
dchecks	array	Проверки обнаружения для создания для правила обнаружения. Поведение параметра: *- required*

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных правил обнаружения, указанных в свойстве `druleids`. Порядок возвращаемых ID совпадает с порядком переданных правил обнаружения.

Примеры

Создание правила обнаружения

Создайте правило обнаружения, чтобы найти машины, на которых запущен агент Zabbix, в локальной сети. Правило должно использовать одну проверку агента Zabbix на порту 10050.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "drule.create",
    "params": {
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "concurrency_max": "10",
        "dchecks": [
            {
                "type": "9",
                "key_": "system.uname",
                "ports": "10050",
                "uniq": "0"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

См. также

- Проверка обнаружения

Источник

CDRule::create() в *ui/include/classes/api/services/CDRule.php*.

drule.delete

Описание

`object drule.delete(array discoveryRuleIds)`

Этот метод позволяет удалять правила обнаружения.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых правил обнаружения.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных правил обнаружения, указанных в свойстве `druleids`.

Примеры

Удаление нескольких правил обнаружения

Удаление двух правил обнаружения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Источник

CDRule::delete() в *ui/include/classes/api/services/CDRule.php*.

drule.get

Описание

`integer/array drule.get(object parameters)`

Метод позволяет получить правила обнаружения в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object) Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dhostids	ID/array	Возвращать только правила обнаружения, которые создали указанные обнаруженные узлы сети.
druleids	ID/array	Возвращать только правила обнаружения с указанными ID.
dserviceids	ID/array	Возвращать только правила обнаружения, которые создали указанные обнаруженные сервисы.
selectDChecks	query	Возвращать свойство dchecks с проверками обнаружения, используемыми правилом обнаружения. Поддерживает count.
selectDHosts	query	Возвращать свойство dhosts с обнаруженными узлами сети, созданными правилом обнаружения. Поддерживает count.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: selectDChecks - результаты будут отсортированы по dcheckid; selectDHosts - результаты будут отсортированы по dhostsid.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: druleid, name.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр countOutput.

Примеры

Получить все правила обнаружения

Получить все настроенные правила обнаружения и проверки обнаружения, которые они используют.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "druleid": "2",
            "proxyid": "0",
            "name": "Local network",
            "iprange": "192.168.3.1-255",
            "delay": "5s",
            "status": "0",
            "concurrency_max": "0",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "7",
                    "druleid": "2",
                    "type": "3",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "21",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                },
                {
                    "dcheckid": "8",
                    "druleid": "2",
                    "type": "4",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "80",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                }
            ]
        },
        {
            "druleid": "6",
            "proxyid": "0",
            "name": "Zabbix agent discovery",
            "iprange": "192.168.1.1-255",
            "delay": "1h",
            "status": "0",
            "concurrency_max": "10",
            "error": "",
```

```
            "dchecks": [
                {
                    "dcheckid": "10",
                    "druleid": "6",
                    "type": "9",
                    "key_": "system.uname",
                    "snmp_community": "",
                    "ports": "10050",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "2",
                    "name_source": "3",
                    "allow_redirect": "0"
                }
            ]
        }
    ],
    "id": 1
}
```


См. также

- Обнаруженный узел сети
- Проверка обнаружения

Источник

CDRule::get() в *ui/include/classes/api/services/CDRule.php*.

drule.update

Описание

`object drule.update(object/array discoveryRules)`

Этот метод позволяет обновлять существующие правила обнаружения.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства правила обнаружения, которые необходимо обновить.

Для каждого правила обнаружения должно быть определено свойство `druleid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам правила обнаружения, метод принимает следующие параметры.

Parameter	Type	Description
dchecks	array	Проверки обнаружения для замены существующих проверок.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID обновленных правил обнаружения под свойством `druleids`.

Примеры

Изменение диапазона IP адресов в правиле обнаружения

Изменение в правила обнаружения диапазона IP адресов на ”192.168.2.1-255”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

См. также

- Проверка обнаружения

Источник

CDRule::update() в *ui/include/classes/api/services/CDRule.php*.

Проблема

Этот класс предназначен для работы с проблемами.

Ссылки на объекты:

- Проблема
  - URL типа медиа
- Тег проблемы

Доступные методы:

- problem.get - получить проблемы

Объект problem

Следующие объекты напрямую связаны с API `problem`.

Проблема

> **Note:**
> Проблемы создаются сервером Zabbix и не могут быть изменены через API.

Объект проблемы имеет следующие свойства.

Property	Type	Description
eventid	ID	ID события проблемы.

Property	Type	Description
source	integer	Тип события проблемы. Возможные значения: 0 - событие, созданное триггером; 3 - внутреннее событие; 4 - событие, созданное при обновлении статуса сервиса.
object	integer	Тип объекта, связанного с событием проблемы. Возможные значения, если `source` установлен в ”event created by a trigger”: 0 - триггер. Возможные значения, если `source` установлен в ”internal event”: 0 - триггер; 4 - элемент данных; 5 - правило LLD. Возможные значения, если `source` установлен в ”event created on service status update”: 6 - сервис.
objectid	ID	ID связанного объекта.
clock	timestamp	Время создания события проблемы.
ns	integer	Наносекунды в момент создания события проблемы.
r_eventid	ID	ID события восстановления.
r_clock	timestamp	Время создания события восстановления.
r_ns	integer	Наносекунды в момент создания события восстановления.
cause_eventid	ID	ID события причины.
correlationid	ID	ID правила корреляции, если это событие было восстановлено глобальным правилом корреляции.
userid	ID	ID пользователя, закрывшего проблему (если проблема была закрыта вручную).
name	string	Имя решенной проблемы.
acknowledged	integer	Состояние подтверждения проблемы. Возможные значения: 0 - не подтверждена; 1 - подтверждена.
severity	integer	Текущая серьезность проблемы. Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.
suppressed	integer	Подавлена ли проблема. Возможные значения: 0 - проблема в нормальном состоянии; 1 - проблема подавлена.
opdata	string	Оперативные данные с раскрытыми макросами.
urls	array	Активные URL-адреса типа media type.

URL типа оповещения

Объект URL типа оповещения имеет следующие свойства.

Property	Type	Description
name	string	Имя URL, определенное для типа оповещения.

Property	Type	Description
url	string	Значение URL, определенное для типа оповещения.

Результаты будут содержать записи только для активных типов оповещения с включенным пунктом меню событий. Макрос, используемый в свойствах, будет раскрыт, но если одно из свойств содержит нераскрытый макрос, оба свойства будут исключены из результатов. Поддерживаемые макросы см. в разделе *Поддерживаемые макросы*.

Тег проблемы

Объект тега проблемы имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега проблемы.
value	string	Значение тега проблемы.

problem.get

Описание

`integer/array problem.get(object parameters)`

Метод позволяет получать проблемы в соответствии с заданными параметрами.

Этот метод предназначен для получения нерешенных проблем. При необходимости также можно дополнительно получать недавно решенные проблемы. Период, определяющий, насколько давно считается "недавно", задается в *Администрирование* > *Общие*. Проблемы, которые были решены раньше этого периода, не сохраняются в таблице проблем. Чтобы получить проблемы, решенные еще раньше, используйте метод `event.get`.

Attention:
Этот метод может возвращать проблемы удаленного объекта, если эти проблемы еще не были удалены housekeeper.

Note:
Этот метод доступен пользователям любого типа. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
eventids	ID/array	Возвращать только проблемы с указанными ID.
groupids	ID/array	Возвращать только проблемы, созданные объектами, которые принадлежат указанным группам узлов сети.
hostids	ID/array	Возвращать только проблемы, созданные объектами, которые принадлежат указанным узлам сети.
objectids	ID/array	Возвращать только проблемы, созданные указанными объектами.
source	integer	Возвращать только проблемы с указанным типом. См. список поддерживаемых типов событий на странице объекта события проблемы problem event object page. По умолчанию: 0 - проблема, созданная триггером.

Parameter	Type	Description
object	integer	Возвращать только проблемы, созданные объектами указанного типа. См. список поддерживаемых типов объектов на странице объекта события проблемы problem event object page. По умолчанию: 0 - триггер.
acknowledged	boolean	Если установлено в `true`, возвращать только подтвержденные проблемы.
action	integer	Возвращать только проблемы, для которых были выполнены указанные действия обновления события. Для нескольких действий используйте сумму любых допустимых значений битовой маски в качестве маски (например, 34 для подтверждения и подавления события).
action_userids	ID/array	Возвращать только проблемы с указанными ID пользователей, выполнивших действия обновления события проблемы.
suppressed	boolean	Если установлено в `true`, возвращать только подавленные проблемы.
symptom	boolean	Если установлено в `true`, возвращать только события проблем-симптомов.
severities	integer/array	Возвращать только проблемы с указанными уровнями серьезности события. Применяется только если object - trigger.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
tags	array	Возвращать только проблемы с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все проблемы. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
recent	boolean	Если установлено в `true`, возвращать недавние проблемы, включая те, которые были решены в пределах `ok_period`, а также любые нерешенные проблемы. Если установлено в `false`, возвращать только нерешенные проблемы.
eventid_from	string	Возвращать только проблемы с ID больше или равным указанному ID.
eventid_till	string	Возвращать только проблемы с ID меньше или равным указанному ID.
time_from	timestamp	Возвращать только проблемы, созданные после указанного времени или в указанное время.
time_till	timestamp	Возвращать только проблемы, созданные до указанного времени или в указанное время.

Parameter	Type	Description
selectAcknowledges	query	Возвращать свойство `acknowledges` с обновлениями проблемы. Обновления проблемы сортируются в обратном хронологическом порядке. Объект обновления проблемы имеет следующие свойства: `acknowledgeid` - (`ID`) ID обновления; `userid` - (`ID`) ID пользователя, который обновил событие; `eventid` - (`ID`) ID обновленного события; `clock` - (`timestamp`) время, когда событие было обновлено; `message` - (`string`) текст сообщения; `action` - (`integer`)тип действия обновления (см. `event.acknowledge`); `old_severity` - (`integer`) уровень серьезности события до этого действия обновления; `new_severity` - (`integer`) уровень серьезности события после этого действия обновления; `suppress_until` - (`timestamp`) время, до которого событие будет подавлено; `taskid` - (`ID`) ID задачи, если текущее событие проходит изменение ранга; `maintenanceid` - (`ID`) ID обслуживания. Поддерживает `count`.
selectTags	query	Возвращать свойство `tags` с тегами проблемы. Формат вывода: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Возвращать свойство `suppression_data` со списком активных обслуживаний и ручных подавлений: `maintenanceid` - (`ID`) ID обслуживания; `userid` - (`ID`) ID пользователя, который подавил проблему; `suppress_until` - (`integer`) время, до которого проблема подавлена.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи - это имена свойств, а значения - либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства с типом данных `text` data type.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `eventid`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(`integer/array`) Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение событий проблем триггера

Получите недавние события от триггера ”15112.”

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectTags": "extend",
        "selectSuppressionData": "extend",
        "objectids": "15112",
        "recent": true,
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "1",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
```

```
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

Получение проблем, подтвержденных указанным пользователем

Получение проблем, подтвержденных пользователем с ID=10

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "    Zabbix   localhost             5    ",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
```

```
    "id": 1
}
```

См. также

- Оповещение
- Элемент данных
- Узел сети
- Правило LLD
- Триггер

Источник

CEvent::get() в *ui/include/classes/api/services/CProblem.php*.

Проверка обнаружения

Этот класс предназначен для работы с проверками обнаружения.

Ссылки на объекты:

- Проверка обнаружения

Доступные методы:

- dcheck.get - получить проверки обнаружения

Объект проверки обнаружения

Следующие объекты напрямую связаны с API `dcheck`.

Проверка обнаружения

Объект проверки обнаружения определяет конкретную проверку, выполняемую правилом сетевого обнаружения. Он имеет следующие свойства.

Property	Type	Description
dcheckid	ID	ID проверки обнаружения.
druleid	ID	ID правила обнаружения, которому принадлежит проверка.
key_	string	Ключ элемента данных (если `type` установлен в ”Zabbix agent”) или OID SNMP (если `type` установлен в ”SNMPv1 agent”, ”SNMPv2 agent” или ”SNMPv3 agent”). Property behavior: - *required* if `type` is set to ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent”, or ”SNMPv3 agent”
ports	string	Один или несколько диапазонов портов для проверки, разделенных запятыми. По умолчанию: 0. Property behavior: - *supported* if `type` is set to ”SSH” (0), ”LDAP” (1), ”SMTP” (2), ”FTP” (3), ”HTTP” (4), ”POP” (5), ”NNTP” (6), ”IMAP” (7), ”TCP” (8), ”Zabbix agent” (9), ”SNMPv1 agent” (10), ”SNMPv2 agent” (11), ”SNMPv3 agent” (13), ”HTTPS” (14), or ”Telnet” (15)
snmp_community	string	SNMP community. Property behavior: - *required* if `type` is set to ”SNMPv1 agent” or ”SNMPv2 agent”
snmpv3_authpassphrase	string	Фраза-пароль аутентификации. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent” and `snmpv3_securitylevel` is set to ”authNoPriv” or ”authPriv”

Property	Type	Description
snmpv3_authprotocol	integer	Протокол аутентификации. Possible values: 0 - *(default)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent” and `snmpv3_securitylevel` is set to ”authNoPriv” or ”authPriv”
snmpv3_contextname	string	Имя контекста SNMPv3. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent”
snmpv3_privpassphrase	string	Фраза-пароль конфиденциальности. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent” and `snmpv3_securitylevel` is set to ”authPriv”
snmpv3_privprotocol	integer	Протокол конфиденциальности. Possible values: 0 - *(default)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent” and `snmpv3_securitylevel` is set to ”authPriv”
snmpv3_securitylevel	string	Уровень безопасности. Possible values: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent”
snmpv3_securityname	string	Имя безопасности. Property behavior: - *supported* if `type` is set to ”SNMPv3 agent”

Property	Type	Description
type	integer	Тип проверки. Possible values: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet. Property behavior: - *required*
uniq	integer	Использовать ли эту проверку как критерий уникальности устройства. Для правила обнаружения может быть настроена только одна уникальная проверка. Possible values: 0 - *(default)* не использовать эту проверку как критерий уникальности; 1 - использовать эту проверку как критерий уникальности. Property behavior: - *supported* if `type` is set to ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent”, or ”SNMPv3 agent”
host_source	integer	Источник имени узла сети. Possible values: 1 - *(default)* DNS; 2 - IP; 3 - значение обнаружения этой проверки.
name_source	integer	Источник видимого имени. Possible values: 0 - *(default)* не указано; 1 - DNS; 2 - IP; 3 - значение обнаружения этой проверки.
allow_redirect	integer	Разрешить ситуацию, когда цель, на которую выполняется ICMP ping, отвечает с другого IP-адреса. Possible values: 0 - *(default)* считать перенаправленные ответы так, как если бы целевой узел сети был недоступен (fail); 1 - считать перенаправленные ответы так, как если бы целевой узел сети был доступен (success). Property behavior: - *supported* if `type` is set to ”ICMP ping”

dcheck.get

Описание

`integer/array dcheck.get(object parameters)`

Метод позволяет получить проверки обнаружения в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
dcheckids	ID/array	Возвращать только проверки обнаружения с указанными ID.
druleids	ID/array	Возвращать только проверки обнаружения, которые принадлежат указанным правилам обнаружения.
dserviceids	ID/array	Возвращать только проверки обнаружения, которые обнаружили указанные обнаруженные службы.
selectDRules	query	Возвращать правила обнаружения, связанные с проверками обнаружения.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `dcheckid`, `druleid`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение проверок обнаружения для правила обнаружения

Получить все проверки обнаружения, используемые правилом обнаружения ”6”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
        }
    ],
    "id": 1
}
```


Источник

CDCheck::get() в *ui/include/classes/api/services/CDCheck.php*.

Прокси

Этот класс предназначен для работы с прокси.

Ссылки на объекты:

- Proxy

Доступные методы:

- proxy.create - создать новые прокси
- proxy.delete - удалить прокси
- proxy.get - получить прокси
- proxy.update - обновить прокси

Объект proxy

Следующие объекты напрямую связаны с API `proxy`.

Прокси

Объект прокси имеет следующие свойства.

Property	Type	Description
proxyid	ID	Идентификатор прокси. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя прокси. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
proxy_groupid	ID	Идентификатор группы прокси. 0 - прокси не назначен ни одной группе прокси.
local_address	string	Адрес для активных агентов. IP-адрес или DNS-имя для подключения. Поведение свойства: - *обязательно*, если `proxy_groupid` не равен 0
local_port	string	Номер локального порта прокси для подключения. Поддерживаются пользовательские макросы. По умолчанию: 10051. Поведение свойства: - *поддерживается*, если `proxy_groupid` не равен 0
operating_mode	integer	Тип прокси. Возможные значения: 0 - активный прокси; 1 - пассивный прокси. Поведение свойства: - *обязательно* для операций создания
description	text	Описание прокси.
lastaccess	timestamp	Время последнего подключения прокси к серверу. Поведение свойства: - *только для чтения*
address	string	IP-адрес или DNS-имя для подключения. Поддерживаются пользовательские макросы. Поведение свойства: - *обязательно*, если режим работы прокси Zabbix - пассивный
port	string	Номер порта для подключения. Поддерживаются пользовательские макросы. По умолчанию: 10051. Поведение свойства: - *поддерживается*, если режим работы прокси Zabbix - пассивный
allowed_addresses	string	IP-адреса или DNS-имена активного прокси Zabbix, разделенные запятыми.
tls_connect	integer	Подключения к прокси. Возможные значения: 1 - *(по умолчанию)* без шифрования; 2 - PSK; 4 - сертификат.
tls_accept	integer	Подключения от прокси. Возможные значения битовой маски: 1 - *(по умолчанию)* без шифрования; 2 - PSK; 4 - сертификат. Это поле является битовой маской; допускается любая сумма возможных значений битовой маски (например, 6 для PSK и сертификата).

Property	Type	Description
tls_issuer	string	Издатель сертификата.
tls_subject	string	Субъект сертификата.
tls_psk_identity	string	Идентификатор PSK; должен быть связан только с одним PSK (в рамках автоматической регистрации, узлов сети и прокси). Не включайте конфиденциальную информацию в идентификатор PSK, поскольку он передается по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK следует использовать. Поведение свойства: - *только для записи* - *обязательно*, если для `tls_connect` установлено значение ”PSK” или `tls_accept` содержит бит ”PSK”
tls_psk	string	Предварительно согласованный ключ (PSK); должен содержать не менее 32 шестнадцатеричных цифр. Поведение свойства: - *только для записи* - *обязательно*, если для `tls_connect` установлено значение ”PSK” или `tls_accept` содержит бит ”PSK”
custom_timeouts	integer	Переопределять ли глобальные тайм-ауты элементов данных на уровне прокси. Возможные значения: 0 - *(по умолчанию)* использовать глобальные настройки; 1 - переопределять тайм-ауты.
timeout_zabbix_agent	string	Тратить на обработку не более `timeout_*` секунд. Принимается количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m). Также принимаются пользовательские макросы. Используется только если для `custom_timeouts` установлено значение ”override timeouts”; в противном случае значение игнорируется. Допустимый диапазон значений: 1-600s. Поведение свойства: - *обязательно*, если для `custom_timeouts` установлено значение ”override timeouts”
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
version	integer	Версия прокси. Трехчастный номер версии Zabbix, в котором для каждой части используются две десятичные цифры, например 60401 для версии 6.4.1, 70002 для версии 7.0.2 и т. д. 0 - неизвестная версия прокси. Поведение свойства: - *только для чтения*

Property	Type	Description
compatibility	integer	Версия прокси по сравнению с версией сервера Zabbix. Возможные значения: 0 - не определено; 1 - текущая версия (прокси и сервер имеют одинаковую основную версию); 2 - устаревшая версия (версия прокси старше версии сервера, но частично поддерживается); 3 - неподдерживаемая версия (версия прокси старше предыдущего выпуска LTS сервера или основная версия сервера старше основной версии прокси). Поведение свойства: - *только для чтения*
state	integer	Состояние прокси. Возможные значения: 0 - неизвестно; 1 - не в сети; 2 - в сети. Поведение свойства: - *только для чтения*

Мониторинг производительности приложений (APM)

Объект APM имеет следующие свойства.

Property	Type	Description
data_collection_status	integer	Определяет, включен ли сбор данных телеметрии на прокси. Возможные значения: 0 - *(по умолчанию)* отключен; 1 - включен.
max_messages_per_second	integer	Максимальное количество сообщений в секунду, которое принимает прокси. Допустимый диапазон значений: от 0 (без ограничений) до 2147483647. Значение по умолчанию: 0. Поведение свойства: - *поддерживается*, если для `data_collection_status` установлено значение ”Enabled”
additional_resource_attributes	array	Массив атрибутов ресурсов, добавляемых к входящим данным телеметрии. Если входящие данные уже содержат атрибут ресурса с таким же именем, вместо указанного здесь сохраняется входящий атрибут. Поведение свойства: - *поддерживается*, если для `data_collection_status` установлено значение ”Enabled”

Атрибуты ресурса

Объект атрибутов ресурса имеет следующие свойства.

Property	Type	Description
key	string	Имя атрибута ресурса. Должно быть уникальным для указанного `signal_type`. Поведение свойства: - *обязательное*
value	string	Значение атрибута ресурса. Может быть пустым.
signal_type	integer	Тип телеметрических данных, к которым применяется атрибут. Возможные значения: 0 - Трассировки; 1 - Метрики; 2 - Журналы. Поведение свойства: - *обязательное*

proxy.create

Описание

`object proxy.create(object/array proxies)`

Этот метод позволяет создавать новые прокси.

> **Note:**
> Этот метод доступен только пользователю типа *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прокси для создания.

Помимо стандартных свойств прокси, метод принимает следующие параметры.

Параметр	Тип	Описание
hosts	array	Узлы сети, которые должны отслеживаться прокси. Если узел сети уже отслеживается другим прокси, он будет переназначен текущему прокси. Узлы сети должны иметь только свойство `hostid`.
apm	object	Настройки мониторинга производительности приложений (APM) прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных прокси в свойстве `proxyids`. Порядок возвращаемых идентификаторов соответствует порядку переданных прокси.

Примеры

Создание активного прокси

Создайте активный прокси "Active proxy" и назначьте ему узел сети для мониторинга.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
```

```
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Создание пассивного прокси

Создайте пассивный прокси "Passive proxy" и назначьте ему два узла сети для мониторинга.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Passive proxy",
        "operating_mode": "1",
        "address": "127.0.0.1",
        "port": "10051",
        "hosts": [
            {
                "hostid": "10192"
            },
            {
                "hostid": "10139"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

Создание прокси и добавление его в группу прокси

Создайте активный прокси "Active proxy" и добавьте его в группу прокси с ID "1".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
```

```
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Создание прокси с шифрованием на основе сертификатов

Создайте активный прокси с шифрованием на основе сертификатов. Прокси настроен на мониторинг узла сети ”10785”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

Создание прокси с включенным APM

Создайте активный прокси «Active proxy» с включенным APM. Прокси принимает до 1000 сообщений в секунду, добавляет атрибут ресурса `deployment.environment` к трассировкам и атрибут ресурса `cloud.region` к метрикам.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
```

```
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "id": 1
}
```

См. также

- Узел сети
- Группа прокси

Источник

CProxy::create() в *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Описание

`object proxy.delete(array proxies)`

Этот метод позволяет удалять прокси.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных прокси в свойстве `proxyids`.

Примеры

Удаление нескольких прокси

Удаление двух прокси.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Источник

CProxy::delete() в *ui/include/classes/api/services/CProxy.php*.

proxy.get

Описание

`integer/array proxy.get(object parameters)`

Метод позволяет получить прокси в соответствии с указанными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.
 Возвращаются только прокси, разрешённые *списком доступа к прокси* групп пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
proxyids	ID/array	Возвращать только прокси с указанными идентификаторами.
proxy_groupids	ID/array	Возвращать только прокси, относящиеся к указанным группам прокси.
selectApm	query	Возвращать свойство `apm` с настройками мониторинга производительности приложений (APM) прокси.
selectAssignedHosts	query	Возвращать свойство `assignedHosts` с узлами сети, назначенными прокси. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, мониторинг которых выполняется прокси. Поддерживает `count`.
selectProxyGroup	query	Возвращать свойство `proxyGroup` с объектом группы прокси.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Описание этих параметров приведено в справочном комментарии.

Параметр	Тип	Описание
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получить все прокси

Получить все настроенные прокси и их интерфейсы.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "11",
            "name": "Active proxy",
            "proxy_groupid": "0",
            "local_address": "",
            "local_port": "10051",
            "operating_mode": "0",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "0",
            "timeout_zabbix_agent": "",
            "timeout_simple_check": "",
            "timeout_snmp_agent": "",
            "timeout_external_check": "",
            "timeout_db_monitor": "",
            "timeout_http_agent": "",
            "timeout_ssh_agent": "",
            "timeout_telnet_agent": "",
```

```
            "timeout_script": "",
            "last_access": "1693391880",
            "version": "70000",
            "compatibility": "1",
            "state": "1"
        },
        {
            "proxyid": "12",
            "name": "Passive proxy",
            "proxy_groupid": "1",
            "local_address": "127.0.0.1",
            "local_port": "10051",
            "operating_mode": "1",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "1",
            "timeout_zabbix_agent": "5s",
            "timeout_simple_check": "5s",
            "timeout_snmp_agent": "5s",
            "timeout_external_check": "5s",
            "timeout_db_monitor": "5s",
            "timeout_http_agent": "5s",
            "timeout_ssh_agent": "5s",
            "timeout_telnet_agent": "5s",
            "timeout_script": "5s",
            "lastaccess": "1693391875",
            "version": "60400",
            "compatibility": "2",
            "state": "2"
        }
    ],
    "id": 1
}
```


См. также

- Узел сети
- Группа прокси

Источник

CProxy::get() в *ui/include/classes/api/services/CProxy.php*.

proxy.update

Описание

`object proxy.update(object/array proxies)`

Этот метод позволяет обновлять существующие прокси.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Свойства прокси, которые необходимо обновить.

Для каждого прокси необходимо определить свойство `proxyid`; все остальные свойства являются необязательными. Будут обновлены только переданные свойства, а все остальные останутся без изменений.

Помимо стандартных свойств прокси, метод принимает следующие параметры.

Parameter	Type	Description
hosts	array	Узлы сети, которые должны отслеживаться прокси. Если узел сети уже отслеживается другим прокси, он будет переназначен текущему прокси. У узлов сети должно быть определено только свойство `hostid`.
apm	object	Настройки мониторинга производительности приложений (APM) прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных прокси в свойстве `proxyids`.

Примеры

Изменение узлов сети, контролируемых прокси

Обновите прокси, чтобы он контролировал два указанных узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Изменение статуса прокси

Измените прокси на активный прокси и переименуйте его в ”Active proxy”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Добавление прокси в группу прокси

Обновите прокси с ID ”5” и добавьте его в группу прокси с ID ”1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

См. также

- Узел сети
- Группа прокси

Источник

CProxy::update() в *ui/include/classes/api/services/CProxy.php*.

Прототип графика

Этот класс предназначен для работы с прототипами графиков.

Ссылки на объекты:

- Прототип графика

Доступные методы:

- graphprototype.create - создать новые прототипы графиков
- graphprototype.delete - удалить прототипы графиков
- graphprototype.get - получить прототипы графиков
- graphprototype.update - обновить прототипы графиков

Объект прототипа графика

Следующие объекты напрямую связаны с API `graphprototype`.

Прототип графика

Объект прототипа графика имеет следующие свойства.

Property	Type	Description
graphid	ID	ID прототипа графика. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
height	integer	Высота прототипа графика в пикселях. Поведение свойства: - *обязательно* для операций создания
name	string	Имя прототипа графика. Поведение свойства: - *обязательно* для операций создания
width	integer	Ширина прототипа графика в пикселях. Поведение свойства: - *обязательно* для операций создания
flags	integer	Происхождение прототипа графика. Возможные значения: 2 - прототип графика; 6 - обнаруженный прототип графика Поведение свойства: - *только для чтения*
graphtype	integer	Тип макета прототипа графика. Возможные значения: 0 - *(по умолчанию)* обычный; 1 - stacked; 2 - pie; 3 - exploded.
percent_left	float	Левый процентиль. По умолчанию: 0.
percent_right	float	Правый процентиль. По умолчанию: 0.
show_3d	integer	Показывать ли обнаруженные круговые и exploded-графики в 3D. Возможные значения: 0 - *(по умолчанию)* показывать в 2D; 1 - показывать в 3D.
show_legend	integer	Показывать ли легенду на обнаруженном графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показать.
show_work_period	integer	Показывать ли рабочее время на обнаруженном графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показать.

Property	Type	Description
templateid	ID	ID родительского прототипа графика шаблона. Поведение свойства: - *только для чтения*
yaxismax	float	Фиксированное максимальное значение для оси Y.
yaxismin	float	Фиксированное минимальное значение для оси Y.
ymax_itemid	ID	ID элемента данных, который используется как максимальное значение для оси Y. Если у пользователя нет доступа к указанному элементу данных, график отображается так, как если бы `ymax_type` был установлен в "calculated".
ymax_type	integer	Метод вычисления максимального значения для оси Y. Возможные значения: 0 - *(по умолчанию)* calculated; 1 - fixed; 2 - item.
ymin_itemid	ID	ID элемента данных, который используется как минимальное значение для оси Y. Если у пользователя нет доступа к указанному элементу данных, график отображается так, как если бы `ymin_type` был установлен в "calculated".
ymin_type	integer	Метод вычисления минимального значения для оси Y. Возможные значения: 0 - *(по умолчанию)* calculated; 1 - fixed; 2 - item.
discover	integer	Статус обнаружения прототипа графика. Возможные значения: 0 - *(по умолчанию)* новые графики будут обнаруживаться; 1 - новые графики не будут обнаруживаться, а существующие графики будут помечены как потерянные.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов графиков с уже существующими. Если не указан, генерируется автоматически. Поведение свойства: - *поддерживается*, если прототип графика принадлежит шаблону

graphprototype.create

Описание

`object graphprototype.create(object/array graphPrototypes)`

Этот метод позволяет создавать новые прототипы графиков.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Прототипы графиков для создания.

Помимо стандартных свойств прототипа графика, метод принимает следующие параметры.

Parameter	Type	Description
gitems	array	Элементы графика, которые нужно создать для прототипов графиков. Элементы графика могут ссылаться как на элементы данных, так и на прототипы элементов данных, но должен присутствовать как минимум один прототип элемента данных. Parameter behavior: *- required*

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных прототипов графиков, указанных в свойстве graphids. Порядок возвращаемых ID совпадает с порядком переданных прототипов графиков.

Примеры

Создание прототипа графика

Создайте прототип графика с двумя элементами данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "                    {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

См. также

- Элемент данных графика

Источник

CGraphPrototype::create() в *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Описание

`object graphprototype.delete(array graphPrototypeIds)`

Этот метод позволяет удалять прототипы графиков.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых прототипов графиков.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных прототипов графиков, указанных в свойстве `graphids`.

Примеры

Удаление нескольких прототипов графиков

Удаление двух прототипов графиков.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Источник

CGraphPrototype::delete() в *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Описание

`integer/array graphprototype.get(object parameters)`

Метод позволяет получать прототипы графиков в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
discoveryids	ID/array	Возвращать только прототипы графиков, которые принадлежат указанным правилам обнаружения.
graphids	ID/array	Возвращать только прототипы графиков с указанными ID.
groupids	ID/array	Возвращать только прототипы графиков, которые принадлежат узлам сети или шаблонам в указанных группах узлов сети или группах шаблонов.
hostids	ID/array	Возвращать только прототипы графиков, которые принадлежат указанным узлам сети.
inherited	boolean	Если задано `true`, возвращать только прототипы графиков, унаследованные от шаблона.
itemids	ID/array	Возвращать только прототипы графиков, которые содержат указанные прототипы элементов данных.
templated	boolean	Если задано `true`, возвращать только прототипы графиков, которые принадлежат шаблонам.
templateids	ID/array	Возвращать только прототипы графиков, которые принадлежат указанным шаблонам.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения прототипа графика. Объект обнаружения прототипа графика связывает обнаруженный прототип графика с прототипом графика, из которого он был обнаружен. Он имеет следующие свойства: `parent_graphid` - `(ID)` ID прототипа графика, на основе которого был создан график; `status` - `(int)` статус обнаружения графика: 0 - *(по умолчанию)* прототип графика обнаружен, 1 - прототип графика больше не обнаружен; `ts_delete` - `(timestamp)` время, когда будет удален прототип графика, который больше не обнаружен.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, к которому принадлежит прототип графика.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, к которому принадлежит прототип графика.
selectGraphItems	query	Возвращать свойство `gitems` с элементами графика, используемыми в прототипе графика.
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, к которым принадлежит прототип графика.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, к которым принадлежит прототип графика.
selectItems	query	Возвращать свойство `items` с элементами данных и прототипами элементов данных, используемыми в прототипе графика.
selectTemplateGroups	query	Возвращать свойство `templategroups` с группами шаблонов, к которым принадлежит прототип графика.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, к которым принадлежит прототип графика.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит прототип графика; `hostid` - ID узла сети, к которому принадлежит прототип графика.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `graphid`, `name`, `graphtype`, `discovered`.

Parameter	Type	Description
countOutput	boolean	Эти параметры описаны в комментарии к справочнику.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение прототипов графиков из правила LLD

Получить все прототипы графиков из правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2"
            "discover": "0"
        }
    ],
```

```
    "id": 1
}
```

См. также

- Правило обнаружения
- Элемент графика
- Элемент данных
- Узел сети
- Группа узлов сети
- Шаблон
- Группа шаблонов

Источник

CGraphPrototype::get() в *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Описание

`object graphprototype.update(object/array graphPrototypes)`

Этот метод позволяет обновлять существующие прототипы графиков.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипа графика, которые необходимо обновить.

Свойство `graphid` должно быть определено для каждого прототипа графика, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Дополнительно к стандартным свойствам прототипа графика, метод принимает следующие параметры.

Parameter	Type	Description
gitems	array	Элементы графика для замены существующих элементов графика. Если у элемента графика определено свойство `gitemid`, он будет обновлен, в противном случае будет создан новый элемент графика.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID обновленных прототипов графиков, указанных в свойстве `graphids`.

Примеры

Изменение размера прототипа графика

Измените размер прототипа графика на 1100 x 400 пикселей.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Источник

CGraphPrototype::update() в *ui/include/classes/api/services/CGraphPrototype.php*.

Прототип правила LLD

Этот класс предназначен для работы с прототипами правил низкоуровневого обнаружения.

Ссылки на объекты:

- Прототип правила LLD
  - HTTP-заголовок
  - Поле HTTP-запроса
- Фильтр прототипа правила LLD
  - Условие фильтра прототипа правила LLD
- Путь макроса прототипа LLD
- Предобработка прототипа правила LLD
- Переопределения прототипа правила LLD
  - Фильтр переопределения прототипа правила LLD
    - Условие фильтра переопределения прототипа правила LLD
  - Операция переопределения прототипа правила LLD
    - Статус операции переопределения прототипа правила LLD
    - Обнаружение операции переопределения прототипа правила LLD
    - Период операции переопределения прототипа правила LLD
    - История операции переопределения прототипа правила LLD
    - Тренды операции переопределения прототипа правила LLD
    - Важность операции переопределения прототипа правила LLD
    - Тег операции переопределения прототипа правила LLD
    - Шаблон операции переопределения прототипа правила LLD
    - Инвентаризация операции переопределения прототипа правила LLD

Доступные методы:

- discoveryruleprototype.create - создать новые прототипы правил LLD
- discoveryruleprototype.delete - удалить прототипы правил LLD
- discoveryruleprototype.get - получить прототипы правил LLD
- discoveryruleprototype.update - обновить прототипы правил LLD

Объект прототипа правила LLD

Следующие объекты напрямую связаны с API `discoveryruleprototype`.

Прототип правила обнаружения низкого уровня

Объект прототипа правила обнаружения низкого уровня имеет следующие свойства.

Свойство	Тип	Описание
itemid	ID	ID прототипа правила обнаружения низкого уровня. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
ruleid	ID	ID родительского правила обнаружения низкого уровня или прототипа правила обнаружения низкого уровня. Поведение свойства: - *только для чтения* - *обязательно* для операций создания
delay	string	Интервал обновления прототипа правила обнаружения низкого уровня. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы. При использовании значение должно состоять из одного макроса. Несколько макросов или макросы в сочетании с текстом не поддерживаются. Гибкие интервалы можно записывать как два макроса, разделённых косой чертой (например, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Пример: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12 Поведение свойства: - *обязательно*, если type имеет значение ”Zabbix agent” (0), ”Simple check” (3), ”Zabbix internal” (5), ”External check” (10), ”Database monitor” (11), ”IPMI agent” (12), ”SSH agent” (13), ”TELNET agent” (14), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”Browser” (22) или если type имеет значение ”Zabbix agent (active)” (7), а key_ не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит прототип правила обнаружения низкого уровня. Поведение свойства: - *константа* - *обязательно* для операций создания
flags	integer	Источник прототипа правила обнаружения низкого уровня. Возможные значения: 3 - прототип правила обнаружения низкого уровня; 7 - обнаруженный прототип правила обнаружения низкого уровня. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
interfaceid	ID	ID интерфейса узла сети прототипа правила обнаружения низкого уровня. Поведение свойства: - *обязательно*, если прототип правила обнаружения низкого уровня принадлежит узлу сети, а `type` имеет значение ”Zabbix agent”, ”IPMI agent”, ”JMX agent” или ”SNMP agent” - *поддерживается*, если прототип правила обнаружения низкого уровня принадлежит узлу сети, а `type` имеет значение ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”
key_	string	Ключ прототипа правила обнаружения низкого уровня. Требуется как минимум один макрос обнаружения низкого уровня. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
name	string	Имя прототипа правила обнаружения низкого уровня. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
type	integer	Тип прототипа правила обнаружения низкого уровня. Возможные значения: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Зависимый элемент данных; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Вложенный. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
url	string	Строка URL. Поддерживает пользовательские макросы, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
allow_traps	integer	Разрешить заполнение значения аналогично элементу данных типа trapper. Возможные значения: 0 - *(по умолчанию)* не принимать входящие данные; 1 - принимать входящие данные. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent”
authtype	integer	Метод аутентификации. Возможные значения, если `type` имеет значение ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если `type` имеет значение ”HTTP agent”: 0 - *(по умолчанию)* отсутствует; 1 - базовая; 2 - NTLM. Поведение свойства: - *поддерживается*, если `type` имеет значение ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)
description	string	Описание прототипа правила обнаружения низкого уровня.
error	string	Текст ошибки, если при обновлении значения прототипа правила обнаружения низкого уровня возникли проблемы. Поведение свойства: - *только для чтения*
follow_redirects	integer	Следовать перенаправлениям ответа при опросе данных. Возможные значения: 0 - не следовать перенаправлениям; 1 - *(по умолчанию)* следовать перенаправлениям. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
headers	array	Массив заголовков, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
http_proxy	string	Строка подключения к прокси HTTP(S). Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Датчик IPMI. Поведение свойства: - *обязательно*, если `type` имеет значение "IPMI agent", а `key_` не имеет значения "ipmi.get" - *поддерживается*, если `type` имеет значение "IPMI agent" - *только для чтения* для унаследованных объектов
jmx_endpoint	string	Пользовательская строка подключения агента JMX. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` имеет значение "JMX agent"
lifetime	string	Период времени, по истечении которого элементы данных, обнаружение которых больше не выполняется, будут удалены. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, `30s`, `1m`, `2h`, `1d`) или пользовательский макрос. По умолчанию: `7d`.
lifetime_type	integer	Сценарий удаления потерянных ресурсов обнаружения низкого уровня. Возможные значения: 0 - *(по умолчанию)* удалить после достижения порогового значения времени существования; 1 - не удалять; 2 - удалить немедленно.
enabled_lifetime	string	Период времени, по истечении которого элементы данных, обнаружение которых больше не выполняется, будут отключены. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, `30s`, `1m`, `2h`, `1d`) или пользовательский макрос. По умолчанию: 0.

Свойство	Тип	Описание
enabled_lifetime_type	integer	Сценарий отключения потерянных ресурсов обнаружения низкого уровня. Возможные значения: 0 - отключить после достижения порогового значения времени существования; 1 - не отключать; 2 - *(по умолчанию)* отключить немедленно.
master_itemid	ID	ID главного элемента данных. Правило обнаружения не может быть главным элементом данных для другого правила обнаружения. Поведение свойства: - *обязательно*, если `type` имеет значение ”Dependent item” - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* сохранить без изменений; 1 - преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
params	string	Дополнительные параметры в зависимости от типа прототипа правила обнаружения низкого уровня: - выполняемый скрипт для правил обнаружения низкого уровня SSH и Telnet; - SQL-запрос для правил обнаружения низкого уровня монитора базы данных; - формула для вычисляемых правил обнаружения низкого уровня; - скрипт для правил обнаружения низкого уровня типа Script и Browser. Поведение свойства: - *обязательно*, если `type` имеет значение ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”Script” или ”Browser”)
parameters	object/array	Дополнительные параметры, если `type` имеет значение ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должно быть уникальным. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Script” или ”Browser” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”JMX agent”, а `username` задано - *поддерживается*, если `type` имеет значение ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)
post_type	integer	Тип тела данных POST, хранящегося в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* необработанные данные; 2 - данные JSON; 3 - данные XML. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”JSON data” или ”XML data” - *поддерживается*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”Raw data” - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
query_fields	array	Массив полей запроса, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
request_method	integer	Тип метода запроса. Возможные значения: 0 - *(по умолчанию)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
retrieve_mode	integer	Какая часть ответа должна сохраняться. Возможные значения, если `request_method` имеет значение "GET", "POST" или "PUT": 0 - *(по умолчанию)* тело; 1 - заголовки; 2 - сохранять тело и заголовки. Возможные значения, если `request_method` имеет значение "HEAD": 1 - заголовки. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
snmp_oid	string	OID SNMP. Поведение свойства: - *обязательно*, если `type` имеет значение "SNMP agent" - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль для файла ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
state	integer	Состояние прототипа правила обнаружения низкого уровня. Возможные значения: 0 - *(по умолчанию)* нормальное; 1 - не поддерживается. Поведение свойства: - *только для чтения*
status	integer	Состояние прототипа правила обнаружения низкого уровня. Возможные значения: 0 - *(по умолчанию)* включённый прототип правила обнаружения низкого уровня; 1 - отключённый прототип правила обнаружения низкого уровня.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Также поддерживает пользовательские макросы как часть списка, разделённого запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
templateid	ID	ID родительского прототипа правила обнаружения низкого уровня шаблона. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса на опрос данных элемента данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-600s. По умолчанию: ”” - использовать настройки прокси или глобальные настройки. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Zabbix agent” (0), ”Simple check” (3), а `key_` не начинается с ”`vmware.`” и ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20), а `snmp_oid` начинается с ”`walk[`” или ”`get[`”, ”Script” (21), ”Browser” (22) - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
trapper_hosts	string	Разрешённые узлы сети. Поведение свойства: - *поддерживается*, если type имеет значение ”Zabbix trapper” или если type имеет значение ”HTTP agent”, а allow_traps имеет значение ”Allow incoming data”. Если свойство не указано, ему автоматически присваивается пользовательский макрос {$TRAPPER.ALLOWED_HOSTS} ( ). {$TRAPPER.ALLOWED_HOSTS} 127.0.0.1,::1. Если свойству присвоена пустая строка, входящие подключения будут запрещены со всех узлов сети.| |username|string|Имя пользователя для аутентификации. [Поведение свойства](/manual/api/reference_commentary#property-behavior): - *обязательно*, еслиtype "SSH agent", "TELNET agent" type "JMX agent", passwordзадано - *поддерживается*, еслитуреимеет значение ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (еслитуреимеет значение ”HTTP agent”)| |uuid|string|Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов правил обнаружения низкого уровня с уже существующими. Генерируется автоматически, если не указан. [Поведение свойства](/manual/api/reference_commentary#property-behavior): - *поддерживается*, если прототип правила обнаружения низкого уровня принадлежит шаблону| |verify_host|integer|Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. [Поведение свойства](/manual/api/reference_commentary#property-behavior): - *поддерживается*, еслитуреимеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов| |verify_peer|integer|Проверять ли подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. [Поведение свойства](/manual/api/reference_commentary#property-

Свойство	Тип	Описание

HTTP-заголовок

Объект заголовка имеет следующие свойства:

Свойство	Тип	Описание
name	string	Имя HTTP-заголовка. Поведение свойства: - *обязательно*
value	string	Значение заголовка. Поведение свойства: - *обязательно*

Поле HTTP-запроса

Объект поля запроса определяет имя и значение, используемые для указания параметра URL. Он имеет следующие свойства:

Свойство	Тип	Описание
name	string	Имя параметра. Поведение свойства: - *обязательно*
value	string	Значение параметра. Поведение свойства: - *обязательно*

Фильтр прототипа правила LLD

Объект фильтра прототипа правила LLD определяет набор условий, которые можно использовать для фильтрации обнаруженных объектов. Он имеет следующие свойства:

Свойство	Тип	Описание
conditions	object/array	Набор условий фильтра, используемых для фильтрации результатов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *обязательно*
evaltype	integer	Метод вычисления условий фильтра. Возможные значения: 0 - И/Или; 1 - И; 2 - Или; 3 - Пользовательское выражение. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
eval_formula	string	Сгенерированное выражение, которое будет использоваться для вычисления условий фильтра. Выражение содержит идентификаторы, которые ссылаются на определённые условия фильтра по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *только для чтения*
formula	string	Определяемое пользователем выражение, используемое для вычисления условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются на определённые условия фильтра по их `formulaid`. Идентификаторы, используемые в выражении, должны в точности соответствовать идентификаторам, определённым в условиях фильтра: ни одно условие не должно остаться неиспользованным или пропущенным. Поведение свойства: - *обязательно*, если `evaltype` установлено в ”custom expression”

Условие фильтра прототипа правила LLD

Объект условия фильтра прототипа правила LLD определяет отдельную проверку, выполняемую для значения макроса LLD. Он имеет следующие свойства:

Свойство	Type	Описание
macro	string	Макрос LLD, для которого выполняется проверка. Property behavior: - *required*
value	string	Значение для сравнения. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. Идентификатор должен быть задан пользователем при изменении условий фильтра, но будет сгенерирован заново при их последующем запросе. Property behavior: - *required* if `evaltype` of объект фильтра прототипа правила LLD is set to ”custom expression”
operator	integer	Оператор условия. Возможные значения: 8 - *(по умолчанию)* соответствует регулярному выражению; 9 - не соответствует регулярному выражению; 12 - существует; 13 - не существует.

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, см. примеры на страницах методов discoveryruleprototype.get и discoveryruleprototype.create.

Путь макроса LLD

Путь макроса LLD имеет следующие свойства:

Свойство	Type	Описание
lld_macro	string	Макрос LLD. Поведение свойства: - *обязательно*
path	string	Селектор значения, которое будет присвоено соответствующему макросу. Поведение свойства: - *обязательно*

Предобработка прототипа правила LLD

Объект предобработки прототипа правила LLD имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип параметра предобработки. Возможные значения: 5 - Регулярное выражение; 11 - XML XPath; 12 - JSONPath; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка наличия ошибки в JSON; 17 - Проверка наличия ошибки в XML; 20 - Отбрасывать неизменившиеся значения с контролем доступности; 21 - JavaScript; 23 - Prometheus в JSON; 24 - CSV в JSON; 25 - Замена; 27 - XML в JSON; 28 - Значение SNMP walk; 29 - SNMP walk в JSON; 30 - Значение SNMP get. Поведение свойства: - *обязательное*
params	string	Дополнительные параметры, используемые параметром предобработки. Несколько параметров разделяются символом новой строки (\n). Поведение свойства: - *обязательное*, если для type установлено значение «Регулярное выражение» (5), «XML XPath» (11), «JSONPath» (12), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Отбрасывать неизменившиеся значения с контролем доступности» (20), «JavaScript» (21), «Prometheus в JSON» (23), «CSV в JSON» (24), «Замена» (25), «Значение SNMP walk» (28), «SNMP walk в JSON» (29) или «Значение SNMP get» (30)

Свойство	Тип	Описание
error_handler	integer	Тип действия, выполняемого в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке задается сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательное*, если для `type` установлено значение «Регулярное выражение» (5), «XML XPath» (11), «JSONPath» (12), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Prometheus в JSON» (23), «CSV в JSON» (24), «XML в JSON» (27), «Значение SNMP walk» (28), «SNMP walk в JSON» (29) или «Значение SNMP get» (30)
error_handler_params	string	Параметры обработчика ошибок. Поведение свойства: - *обязательное*, если для `error_handler` установлено значение «Установить пользовательское значение» или «Установить пользовательское сообщение об ошибке»

Для каждого типа предобработки поддерживаются следующие параметры и обработчики ошибок.

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
5	Регулярное выражение	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Соответствует регулярному выражению	pattern[1]			0, 1, 2, 3
15	Не соответствует регулярному выражению	pattern[1]			0, 1, 2, 3
16	Проверка наличия ошибки в JSON	path[3]			0, 1, 2, 3
17	Проверка наличия ошибки в XML	path[3]			0, 1, 2, 3
20	Отбрасывать неизменившиеся значения с контролем доступности	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus в JSON	pattern^5, 6^			0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
24	CSV в JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Замена	search string[2]	replacement[2]		
27	XML в JSON				0, 1, 2, 3
28	Значение SNMP walk	OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS		0, 1, 2, 3
29	SNMP walk в JSON[7]	Имя поля[2]	Префикс OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS	0, 1, 2, 3
30	Значение SNMP get	Формат: 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS			0, 1, 2, 3

[1] регулярное выражение
 [2] строка
 [3] JSONPath или XML XPath
 [4] положительное целое число (с поддержкой суффиксов времени, например 30s, 1m, 2h, 1d)
 [5] пользовательский макрос
 [6] шаблон Prometheus, соответствующий синтаксису: `<metric name>{<label name>="<label value>", ...} == <value>`. Каждый компонент шаблона Prometheus (метрика, имя метки, значение метки и значение метрики) может быть пользовательским макросом.
 [7] Поддерживает несколько записей «Имя поля,Префикс OID,Формат», разделенных символом новой строки.

Переопределения прототипов правил LLD

Объект переопределений прототипов правил LLD определяет набор правил (фильтров, условий и операций), которые используются для переопределения свойств различных объектов прототипов. Он имеет следующие свойства:

Свойство	Type	Описание
name	string	Уникальное имя переопределения. Поведение свойства: - *обязательно*
step	integer	Уникальный порядковый номер переопределения. Поведение свойства: - *обязательно*
stop	integer	Остановить обработку следующих переопределений при совпадении. Возможные значения: 0 - *(по умолчанию)* не останавливать обработку переопределений; 1 - остановить обработку переопределений, если фильтр совпадает.

Свойство	Type	Описание
filter	object	Фильтр переопределения.
operations	object/array	Операции переопределения.

Фильтр переопределения прототипа правила LLD

Объект фильтра переопределения прототипа правила LLD определяет набор условий, при совпадении которых с обнаруженным объектом применяется переопределение. Он имеет следующие свойства:

Свойство	Тип	Описание
conditions	object/array	Набор условий фильтра переопределения, используемых для сопоставления с обнаруженными объектами. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *обязательно*
evaltype	integer	Метод вычисления условий фильтра переопределения. Возможные значения: 0 - И/Или; 1 - И; 2 - Или; 3 - Пользовательское выражение. Поведение свойства: - *обязательно*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для вычисления условий фильтра переопределения. Выражение содержит идентификаторы, которые ссылаются на конкретные условия фильтра переопределения по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *только для чтения*
formula	string	Определяемое пользователем выражение, используемое для вычисления условий фильтров переопределения с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются на конкретные условия фильтра переопределения по их `formulaid`. Идентификаторы, используемые в выражении, должны в точности соответствовать тем, которые определены в условиях фильтра переопределения: ни одно условие не должно остаться неиспользованным или пропущенным. Поведение свойства: - *обязательно*, если `evaltype` установлен в ”custom expression”

Условие фильтра переопределения прототипа правила LLD

Объект условия фильтра переопределения прототипа правила LLD определяет отдельную проверку, выполняемую для значения макроса LLD. Он имеет следующие свойства:

Свойство	Type	Описание
macro	string	Макрос LLD, для которого выполняется проверка. Property behavior: - *required*

Свойство	Type	Описание
value	string	Значение для сравнения. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Произвольный уникальный идентификатор, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. Идентификатор должен быть определён пользователем при изменении условий фильтра, но будет сгенерирован заново при их последующем запросе. Property behavior: - *required* if `evaltype` of объект фильтра переопределения прототипа правила LLD is set to ”custom expression”
operator	integer	Оператор условия. Возможные значения: 8 - *(по умолчанию)* соответствует регулярному выражению; 9 - не соответствует регулярному выражению; 12 - существует; 13 - не существует.

Операция переопределения прототипа правила LLD

Операция переопределения прототипа правила LLD представляет собой комбинацию условий и действий, которые выполняются для объекта прототипа. Она имеет следующие свойства:

Свойство	Type	Описание
operationobject	integer	Тип обнаруженного объекта, для которого выполняется действие. Возможные значения: 0 - Прототип элемента данных; 1 - Прототип триггера; 2 - Прототип графика; 3 - Прототип узла сети. Поведение свойства: - *обязательно*
operator	integer	Оператор условия переопределения. Возможные значения: 0 - *(по умолчанию)* равно; 1 - не равно; 2 - содержит; 3 - не содержит; 8 - соответствует; 9 - не соответствует.
value	string	Шаблон для сопоставления с именем прототипа элемента данных, триггера, графика или узла сети в зависимости от выбранного объекта.
opstatus	object	Объект статуса операции переопределения для объектов прототипа элемента данных, триггера и узла сети.
opdiscover	object	Объект статуса обнаружения операции переопределения (для всех типов объектов).
opperiod	object	Объект периода операции переопределения (интервала обновления) для объекта прототипа элемента данных.
ophistory	object	Объект истории операции переопределения для объекта прототипа элемента данных.
optrends	object	Объект трендов операции переопределения для объекта прототипа элемента данных.

Свойство	Type	Описание
opseverity	object	Объект важности операции переопределения для объекта прототипа триггера.
optag	object/array	Объект тега операции переопределения для объектов прототипа триггера и узла сети.
optemplate	object/array	Объект шаблона операции переопределения для объекта прототипа узла сети.
opinventory	object	Объект инвентаризации операции переопределения для объекта прототипа узла сети.

Статус операции переопределения прототипа правила LLD

Статус операции переопределения прототипа правила LLD, который устанавливается для обнаруженного объекта. Он имеет следующие свойства:

Свойство	Type	Описание
status	integer	Переопределяет статус для выбранного объекта. Возможные значения: 0 - Создать включённым; 1 - Создать отключённым. Поведение свойства: - *обязательно*

Операция переопределения обнаружения для прототипа правила LLD

Статус обнаружения операции переопределения для прототипа правила LLD, который устанавливается для обнаруженного объекта. Имеет следующие свойства:

Свойство	Type	Описание
discover	integer	Переопределяет статус обнаружения для выбранного объекта. Возможные значения: 0 - Да, продолжать обнаружение объектов; 1 - Нет, новые объекты не будут обнаруживаться, а существующие будут помечены как потерянные. Поведение свойства: - *обязательно*

Период операции переопределения прототипа правила LLD

Период операции переопределения прототипа правила LLD — это значение интервала обновления, заданное для обнаруженного элемента данных. Он имеет следующие свойства:

Property	Type	Description
delay	string	Переопределяет интервал обновления прототипа элемента данных. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой комбинацию гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы или макросы LLD. При использовании значение должно состоять из одного макроса. Несколько макросов или макросы, объединённые с текстом, не поддерживаются. Гибкие интервалы можно записывать как два макроса, разделённых косой чертой (например, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Пример: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Поведение свойства: - *обязательное*

История операции переопределения прототипа правила LLD

Значение истории операции переопределения прототипа правила LLD, которое устанавливается для обнаруженного элемента данных. Имеет следующие свойства:

Свойство	Type	Описание
history	string	Переопределяет историю прототипа элемента данных, то есть единицу времени, в течение которой должны храниться исторические данные. Также принимает пользовательский макрос и макрос LLD. Поведение свойства: - *обязательно*

Тенденции операции переопределения прототипа правила LLD

Значение тенденций операции переопределения прототипа правила LLD, которое задаётся для обнаруженного элемента данных. Оно имеет следующие свойства:

Свойство	Type	Описание
trends	string	Переопределяет тенденции прототипа элемента данных, то есть единицу времени, в течение которой данные тенденций должны храниться. Также принимает пользовательский макрос и макрос LLD. Поведение свойства: - *обязательно*

Значение важности операции переопределения прототипа правила LLD

Значение важности операции переопределения прототипа правила LLD, которое устанавливается для обнаруженного триггера. Имеет следующие свойства:

Свойство	Тип	Описание
severity	integer	Переопределяет важность прототипа триггера. Возможные значения: 0 - *(по умолчанию)* не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа. Поведение свойства: - *обязательно*

Тег операции переопределения прототипа правила LLD

Объект тега операции переопределения прототипа правила LLD содержит имя и значение тега, которые задаются для обнаруженного объекта. Он имеет следующие свойства:

Свойство	Type	Описание
tag	string	Новое имя тега. Поведение свойства: - *обязательно*
value	string	Новое значение тега.

Шаблон операции переопределения прототипа правила LLD

Объект шаблона операции переопределения прототипа правила LLD, который связан с обнаруженным узлом сети. Он имеет следующие свойства:

Свойство	Type	Описание
templateid	ID	Переопределяет шаблон среди связанных шаблонов прототипа узла сети. Поведение свойства: - *обязательно*

Значение режима инвентаризации операции переопределения прототипа правила LLD

Значение режима инвентаризации операции переопределения прототипа правила LLD, которое устанавливается для обнаруженного узла сети. Имеет следующие свойства:

Свойство	Type	Описание
inventory_mode	integer	Переопределяет режим инвентаризации прототипа узла сети. Возможные значения: -1 - отключено; 0 - *(по умолчанию)* вручную; 1 - автоматически. Поведение свойства: - *обязательно*

discoveryruleprototype.create

Описание

```
object discoveryruleprototype.create(object/array lldRules)
```

Этот метод позволяет создавать новые прототипы правил LLD.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототипы правил LLD для создания.

В дополнение к стандартным свойствам прототипа правила LLD, метод принимает следующие параметры.

Параметр	Тип	Описание
filter	object	Фильтр прототипа правила LLD для правила LLD.
preprocessing	object/array	Параметры предобработки прототипа правила LLD.
lld_macro_paths	object/array	Параметры lld_macro_path прототипа правила LLD.
overrides	object/array	Параметры переопределений прототипа правила LLD.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных прототипов правил LLD в свойстве `itemids`. Порядок возвращаемых идентификаторов соответствует порядку переданных прототипов правил LLD.

Примеры

Создание прототипа правила LLD

Создайте прототип правила LLD (Тип: Nested) для обнаружения табличных пространств в экземпляре базы данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

См. также

- Фильтр прототипа правила LLD
- Пути макросов LLD
- Предобработка прототипа правила LLD

Исходный код

CDiscoveryRulePrototype::create() в *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Описание

`object discoveryruleprototype.delete(array lldRuleIds)`

Этот метод позволяет удалять прототипы правил LLD.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы прототипов правил LLD, которые необходимо удалить.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых прототипов правил LLD в свойстве `ruleids`.

Примеры

Удаление нескольких прототипов правил LLD

Удалите два прототипа правил LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

Исходный код

CDiscoveryRulePrototype::delete() в *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Описание

`integer/array discoveryruleprototype.get(object parameters)`

Метод позволяет получать прототипы правил LLD в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешение на вызов метода может быть отозвано в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
discoveryids	ID/array	Возвращать только прототипы правил LLD, принадлежащие указанным правилам LLD или прототипам правил LLD.
itemids	ID/array	Возвращать только прототипы правил LLD с указанными ID.
groupids	ID/array	Возвращать только прототипы правил LLD, принадлежащие узлам сети из указанных групп.
hostids	ID/array	Возвращать только прототипы правил LLD, принадлежащие указанным узлам сети.
inherited	boolean	Если установлено в `true`, возвращать только прототипы правил LLD, унаследованные от шаблона.
interfaceids	ID/array	Возвращать только прототипы правил LLD, использующие указанные интерфейсы узла сети.
monitored	boolean	Если установлено в `true`, возвращать только включённые прототипы правил LLD, принадлежащие отслеживаемым узлам сети.
templated	boolean	Если установлено в `true`, возвращать только прототипы правил LLD, принадлежащие шаблонам.
templateids	ID/array	Возвращать только прототипы правил LLD, принадлежащие указанным шаблонам.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с родительским правилом LLD для прототипа правила LLD. Будет возвращён пустой ответ, если родительский объект является прототипом правила обнаружения.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD для прототипа правила LLD. Будет возвращён пустой ответ, если родительский объект является правилом обнаружения (не прототипом).
selectDiscoveryRulePrototypes	query	Возвращать свойство `discoveryRulePrototypes` с дочерними прототипами правил LLD, принадлежащими прототипу правила LLD. Поддерживает `count`.
selectFilter	query	Возвращать свойство `filter` с данными фильтра, используемого прототипом правила LLD.
selectGraphs	query	Возвращает свойство `graphs` с прототипами графиков, принадлежащими прототипу правила LLD. Поддерживает `count`.
selectHostPrototypes	query	Возвращать свойство `hostPrototypes` с прототипами узлов сети, принадлежащими прототипу правила LLD. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, которым принадлежит прототип правила LLD.
selectItems	query	Возвращать свойство `items` с прототипами элементов данных, принадлежащими прототипу правила LLD. Поддерживает `count`.
selectTriggers	query	Возвращать свойство `triggers` с прототипами триггеров, принадлежащими прототипу правила LLD. Поддерживает `count`.
selectLLDMacroPaths	query	Возвращать свойство `lld_macro_paths` со списком макросов LLD и путей к значениям, назначенным каждому соответствующему макросу.
selectPreprocessing	query	Возвращать свойство `preprocessing` с параметрами предобработки прототипа правила LLD.

Параметр	Тип	Описание
selectOverrides	query	Возвращать свойство `lld_rule_overrides` со списком фильтров переопределения, условий и операций, выполняемых над объектами-прототипами.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства типа данных `text` data type. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, которому принадлежит прототип правила LLD.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение прототипов правил обнаружения с узла сети

Получите все прототипы правил обнаружения для указанного ID узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
```

```
            "snmp_oid": "",
            "hostid": "10084",
            "name": "                        {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


См. также

- Прототип графика
- Узел сети
- Прототип элемента данных
- Фильтр прототипа правила LLD

- Прототип триггера

Исходный код

CDiscoveryRulePrototype::get() в *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Описание

`object discoveryruleprototype.update(object/array lldRules)`

Этот метод позволяет обновлять существующие прототипы правил LLD.

Обратите внимание, что обновление уже обнаруженных прототипов ограничено.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` свойства прототипов правил LLD, которые необходимо обновить.

Свойство `itemid` должно быть определено для каждого прототипа правила LLD, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прототипа правила LLD, метод принимает следующие параметры.

Параметр	Type	Описание
filter	object	Фильтр прототипа правила LLD для замены существующего фильтра.
preprocessing	object/array	Параметры предобработки прототипа правила LLD для замены существующих параметров предобработки. Поведение параметра: - *только для чтения* для унаследованных объектов
lld_macro_paths	object/array	Параметры lld_macro_path прототипа правила LLD для замены существующих параметров lld_macro_path. Поведение параметра: - *только для чтения* для унаследованных объектов
overrides	object/array	Параметры переопределений прототипа правила LLD для замены существующих параметров переопределений. Поведение параметра: - *только для чтения* для унаследованных объектов

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых прототипов правил LLD в свойстве `itemids`.

Примеры

Обновление параметров предобработки прототипа правила LLD

Обновите прототип правила LLD с правилом предобработки JSONPath. Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
```

```
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Исходный код

CDiscoveryRulePrototype::update() в *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

Прототип триггера

Этот класс предназначен для работы с прототипами триггеров.

Ссылки на объекты:

- Прототип триггера
- Тег прототипа триггера

Доступные методы:

- triggerprototype.create - создать новые прототипы триггеров
- triggerprototype.delete - удалить прототипы триггеров
- triggerprototype.get - получить прототипы триггеров
- triggerprototype.update - обновить прототипы триггеров

Объект прототипа триггера

Следующие объекты напрямую связаны с API `triggerprototype`.

Прототип триггера

Объект прототипа триггера имеет следующие свойства.

Property	Type	Description
triggerid	ID	ID прототипа триггера. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
description	string	Имя прототипа триггера. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
expression	string	Выражение триггера. Должно содержать как минимум один прототип элемента данных. Прототипы элементов данных со значением `value_type`, равным 5 (binary) или 6 (JSON), не могут использоваться в выражениях триггера. Поведение свойства: - *обязательно* для операций создания
event_name	string	Имя события, создаваемого триггером.
opdata	string	Оперативные данные.
comments	string	Дополнительные комментарии к прототипу триггера.
priority	integer	Серьезность прототипа триггера. Возможные значения: 0 - *(по умолчанию)* не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.
status	integer	Включен или отключен прототип триггера. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
flags	integer	Происхождение прототипа триггера. Возможные значения: 2 - прототип триггера; 6 - обнаруженный прототип триггера. Поведение свойства: - *только для чтения*
templateid	ID	ID родительского прототипа триггера шаблона. Поведение свойства: - *только для чтения*
type	integer	Может ли прототип триггера генерировать несколько событий проблемы. Возможные значения: 0 - *(по умолчанию)* не генерировать несколько событий; 1 - генерировать несколько событий.
url	string	URL, связанный с прототипом триггера.
url_name	string	Метка для URL, связанного с прототипом триггера.
recovery_mode	integer	Режим генерации события OK. Возможные значения: 0 - *(по умолчанию)* выражение; 1 - выражение восстановления; 2 - нет.
recovery_expression	string	Выражение восстановления триггера. Должно содержать как минимум один прототип элемента данных. Прототипы элементов данных со значением `value_type`, равным 5 (binary) или 6 (JSON), не могут использоваться в выражениях восстановления триггера.

Property	Type	Description
correlation_mode	integer	Закрытие события OK. Возможные значения: 0 - *(по умолчанию)* все проблемы; 1 - все проблемы, если значения тегов совпадают.
correlation_tag	string	Тег для сопоставления.
manual_close	integer	Разрешить ручное закрытие. Возможные значения: 0 - *(по умолчанию)* нет; 1 - да.
discover	integer	Статус обнаружения прототипа триггера. Возможные значения: 0 - *(по умолчанию)* будут обнаруживаться новые триггеры; 1 - новые триггеры не будут обнаруживаться, а существующие триггеры будут помечены как потерянные.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов триггеров с уже существующими. Если не указан, генерируется автоматически. Поведение свойства: - *поддерживается*, если прототип триггера принадлежит шаблону

Тег прототипа триггера

Объект тега прототипа триггера имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега прототипа триггера. Property behavior: - *required*
value	string	Значение тега прототипа триггера. Property behavior: - *read-only*.
object	integer	Тип объекта, от которого был унаследован тег. Possible values: 0 - шаблон; 1 - узел сети; 3 - элемент данных; 4 - прототип элемента данных. Property behavior: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Property behavior: - *read-only*.

triggerprototype.create

Описание

`object triggerprototype.create(object/array triggerPrototypes)`

Этот метод позволяет создавать новые прототипы триггеров.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототипы триггеров для создания.

Помимо стандартных свойств прототипа триггера метод принимает следующие параметры.

Parameter	Type	Description
dependencies	array	Триггеры и прототипы триггеров, от которых зависит прототип триггера. У триггеров должно быть определено только свойство `triggerid`.
tags	array	Теги прототипа триггера.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных прототипов триггеров в свойстве `triggerids`. Порядок возвращаемых идентификаторов соответствует порядку переданных прототипов триггеров.

Примеры

Создание прототипа триггера

Создайте прототип триггера для обнаружения ситуации, когда в файловой системе остается менее 20% свободного дискового пространства.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "                              20%       {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Источник

CTriggerPrototype::create() в *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

Описание

`object triggerprototype.delete(array triggerPrototypeIds)`

Этот метод позволяет удалять прототипы триггеров.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` ID прототипов триггеров для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных прототипов триггеров в свойстве `triggerids`.

Примеры

Удаление нескольких прототипов триггеров

Удаление двух прототипов триггеров.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Источник

CTriggerPrototype::delete() в *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Описание

`integer/array triggerprototype.get(object parameters)`

Метод позволяет получать прототипы триггеров в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object) Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
active	flag	Возвращать только включенные прототипы триггеров, которые принадлежат мониторируемым узлам сети.
discoveryids	ID/array	Возвращать только прототипы триггеров, которые принадлежат указанным правилам LLD.
functions	string/array	Возвращать только триггеры, которые используют указанные функции. См. страницу Поддерживаемые функции для списка поддерживаемых функций.
group	string	Возвращать только прототипы триггеров, которые принадлежат узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
groupids	ID/array	Возвращать только прототипы триггеров, которые принадлежат узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
host	string	Возвращать только прототипы триггеров, которые принадлежат узлам сети с указанным именем.
hostids	ID/array	Возвращать только прототипы триггеров, которые принадлежат указанным узлам сети.
inherited	boolean	Если установлено значение `true`, возвращать только прототипы триггеров, унаследованные от шаблона.
maintenance	boolean	Если установлено значение `true`, возвращать только включенные прототипы триггеров, которые принадлежат узлам сети в обслуживании.
min_severity	integer	Возвращать только прототипы триггеров с уровнем серьезности не ниже указанного.
monitored	flag	Возвращать только включенные прототипы триггеров, которые принадлежат мониторируемым узлам сети и содержат только включенные элементы данных.
templated	boolean	Если установлено значение `true`, возвращать только прототипы триггеров, которые принадлежат шаблонам.
templateids	ID/array	Возвращать только прототипы триггеров, которые принадлежат указанным шаблонам.
triggerids	ID/array	Возвращать только прототипы триггеров с указанными ID.
expandExpression	flag	Разворачивать функции и макросы в выражении триггера.
selectDependencies	query	Возвращать прототипы триггеров и триггеры, от которых зависит прототип триггера, в свойстве `dependencies`.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения прототипа триггера. Объект обнаружения прототипа триггера связывает обнаруженный прототип триггера с прототипом триггера, из которого он был обнаружен. Он имеет следующие свойства: `parent_triggerid` - (`ID`) ID прототипа триггера, из которого был создан прототип триггера; `status` - (`int`) статус обнаружения прототипа триггера: 0 - *(по умолчанию)* прототип триггера обнаружен, 1 - прототип триггера больше не обнаружен; `ts_delete` - (`timestamp`) время, когда прототип триггера, который больше не обнаружен, будет удален; `ts_disable` - (`timestamp`) время, когда прототип триггера, который больше не обнаружен, будет отключен; `disable_source` - (`int`) признак того, был ли прототип триггера отключен правилом LLD или вручную: 0 - *(по умолчанию)* отключен автоматически, 1 - отключен правилом LLD.
selectDiscoveryRule	query	Возвращать правило LLD, к которому принадлежит прототип триггера, в свойстве `discoveryRule`.

Parameter	Type	Description
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, к которому принадлежит прототип триггера.
selectFunctions	query	Возвращать функции, используемые в прототипе триггера, в свойстве `functions`. Объекты функции представляют функции, используемые в выражении триггера, и имеют следующие свойства: `functionid` - (ID) ID функции; `itemid` - (ID) ID элемента данных, используемого в функции; `function` - (string) имя функции; `parameter` - (string) параметр, переданный функции. Параметр запроса заменяется символом $ в возвращаемой строке.
selectHostGroups	query	Возвращать группы узлов сети, к которым принадлежит прототип триггера, в свойстве `hostgroups`.
selectHosts	query	Возвращать узлы сети, к которым принадлежит прототип триггера, в свойстве `hosts`.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, унаследованными от шаблонов, связанных шаблонов и узлов сети, а также от прототипов элементов данных, на которые ссылаются выражения триггера или восстановления.
selectItems	query	Возвращать элементы данных и прототипы элементов данных, используемые прототипом триггера, в свойстве `items`.
selectTags	query	Возвращать теги прототипа триггера в свойстве `tags`.
selectTemplateGroups	query	Возвращать группы шаблонов, к которым принадлежит прототип триггера, в свойстве `templategroups`.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит прототип триггера; `hostid` - ID узла сети, к которому принадлежит прототип триггера.
limitSelects	integer	Ограничивает число записей, возвращаемых подвыборами. Применяется к следующим подвыборам: `selectHosts` - результаты будут отсортированы по `host`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(integer/array) Возвращает либо:

- массив объектов;
- количество полученных объектов, если использован параметр countOutput.

Примеры

Получение прототипов триггеров из правила LLD

Получите все прототипы триггеров и их функции из правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
```

```
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


Получение конкретного прототипа триггера с тегами

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "                    20%       {#FSNAME}",
            "tags": [
```

```
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Правило обнаружения
- Элемент данных
- Узел сети
- Группа узлов сети
- Группа шаблонов

Источник

CTriggerPrototype::get() в *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Описание

`object triggerprototype.update(object/array triggerPrototypes)`

Этот метод позволяет обновлять существующие прототипы триггеров.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипа триггера, которые необходимо обновить.

Для каждого прототипа триггера должно быть определено свойство `triggerid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Кроме стандартных свойств прототипа триггера, метод принимает следующие параметры.

Parameter	Type	Description
dependencies	array	Триггеры и прототипы триггеров, от которых зависит прототип триггера. У триггеров должно быть определено только свойство `triggerid`.
tags	array	Теги прототипа триггера.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных прототипов триггеров в свойстве `triggerids`.

Примеры

Включение прототипа триггера

Чтобы включить прототип триггера, установите для его статуса значение ”0”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Замена тегов прототипа триггера

Заменяет теги для одного прототипа триггера.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Источник

CTriggerPrototype::update() в *ui/include/classes/api/services/CTriggerPrototype.php*.

Прототип узла сети

Этот класс предназначен для работы с прототипами узлов сети.

Ссылки на объекты:

- Прототип узла сети
- Связь группы
- Прототип группы
- Тег прототипа узла сети
- Пользовательский интерфейс
  - Сведения о пользовательском интерфейсе

Доступные методы:

- hostprototype.create - создать новые прототипы узлов сети
- hostprototype.delete - удалить прототипы узлов сети
- hostprototype.get - получить прототипы узлов сети
- hostprototype.update - обновить прототипы узлов сети

Объект прототипа узла сети

Следующие объекты напрямую связаны с API `hostprototype`.

Прототип узла сети

Объект прототипа узла сети имеет следующие свойства.

Property	Type	Description
hostid	ID	ID прототипа узла сети. Property behavior: - *только для чтения* - *обязательно* для операций обновления
host	string	Техническое имя прототипа узла сети. Property behavior: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
name	string	Отображаемое имя прототипа узла сети. По умолчанию: значение свойства `host`. Property behavior: - *только для чтения* для унаследованных объектов
status	integer	Статус прототипа узла сети. Возможные значения: 0 - *(по умолчанию)* контролируемый узел сети; 1 - неконтролируемый узел сети.
flags	integer	Происхождение прототипа узла сети. Возможные значения: 2 - прототип узла сети; 6 - обнаруженный прототип узла сети Property behavior: - *только для чтения*
inventory_mode	integer	Режим заполнения инвентарных данных узла сети. Возможные значения: -1 - *(по умолчанию)* отключено; 0 - вручную; 1 - автоматически.
templateid	ID	ID родительского шаблона прототипа узла сети. Property behavior: - *только для чтения*

Property	Type	Description
discover	integer	Статус обнаружения прототипа узла сети. Возможные значения: 0 - *(по умолчанию)* новые узлы сети будут обнаруживаться; 1 - новые узлы сети не будут обнаруживаться, а существующие узлы сети будут помечены как потерянные.
custom_interfaces	integer	Источник пользовательских интерфейсов для узлов сети, созданных по прототипу узла сети. Возможные значения: 0 - *(по умолчанию)* наследовать интерфейсы от родительского узла сети; 1 - использовать пользовательские интерфейсы прототипов узлов сети. Property behavior: - *только для чтения* для унаследованных объектов
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов узлов сети с уже существующими. Если не указан, генерируется автоматически. Property behavior: - *поддерживается*, если прототип узла сети принадлежит шаблону

Связь группы

Объект связи группы связывает прототип узла сети с группой узлов сети. Он имеет следующие свойства.

Property	Type	Description
groupid	ID	ID группы узлов сети. Property behavior: - *required*

Прототип группы

Объект прототипа группы определяет группу, которая будет создана для обнаруженного узла сети. Он имеет следующие свойства.

Property	Type	Description
group_prototypeid	ID	ID прототипа группы. Property behavior: - *read-only*
name	string	Имя прототипа группы. Property behavior: - *required* for create operations

Тег прототипа узла сети

Объект тега прототипа узла сети имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега прототипа узла сети. Property behavior: - *required*

Property	Type	Description
value	string	Значение тега прототипа узла сети. Property behavior: - *read-only*.
object	integer	Тип объекта, от которого был унаследован тег. Possible values: 0 - шаблон. Property behavior: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Property behavior: - *read-only*.

Пользовательский интерфейс

Пользовательские интерфейсы поддерживаются, если `custom_interfaces` объекта прототип узла сети установлен в значение ”use host prototypes custom interfaces”. Объект пользовательского интерфейса имеет следующие свойства.

Property	Type	Description
type	integer	Тип интерфейса. Возможные значения: 1 - Агент; 2 - SNMP; 3 - IPMI; 4 - JMX. Поведение свойства: - *required*
useip	integer	Следует ли выполнять подключение через IP. Возможные значения: 0 - подключаться, используя DNS-имя узла сети; 1 - подключаться, используя IP-адрес узла сети. Поведение свойства: - *required*
ip	string	IP-адрес, используемый интерфейсом. Может содержать макросы. Поведение свойства: - *required* if `useip` is set to ”connect using host IP address”
dns	string	DNS-имя, используемое интерфейсом. Может содержать макросы. Поведение свойства: - *required* if `useip` is set to ”connect using host DNS name”

Property	Type	Description
port	string	Номер порта, используемый интерфейсом. Может содержать пользовательские макросы и макросы LLD. Поведение свойства: - *required*
main	integer	Используется ли интерфейс по умолчанию на узле сети. Только один интерфейс определенного типа может быть задан как интерфейс по умолчанию на узле сети. Возможные значения: 0 - не по умолчанию; 1 - по умолчанию. Поведение свойства: - *required*
details	object	Дополнительный объект для сведений о пользовательском интерфейсе. Поведение свойства: - *required* if `type` is set to ”SNMP”

Детали пользовательского интерфейса

Объект details имеет следующие свойства.

Property	Type	Description
version	integer	Версия интерфейса SNMP. Возможные значения: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Поведение свойства: - *required*
bulk	integer	Использовать ли массовые SNMP-запросы. Возможные значения: 0 - не использовать массовые запросы; 1 - *(по умолчанию)* - использовать массовые запросы.
community	string	SNMP community. Поведение свойства: - *required* если `version` установлено в ”SNMPv1” или ”SNMPv2c”
max_repetitions	integer	Максимальное значение повторений для нативных массовых SNMP-запросов (GetBulkRequest-PDU). Используется только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. По умолчанию: 10.

Property	Type	Description
securityname	string	Имя безопасности SNMPv3. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3”
securitylevel	integer	Уровень безопасности SNMPv3. Возможные значения: 0 - *(по умолчанию)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3”
authpassphrase	string	Фраза-пароль аутентификации SNMPv3. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authNoPriv” или ”authPriv”
privpassphrase	string	Фраза-пароль конфиденциальности SNMPv3. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authPriv”
authprotocol	integer	Протокол аутентификации SNMPv3. Возможные значения: 0 - *(по умолчанию)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authNoPriv” или ”authPriv”
privprotocol	integer	Протокол конфиденциальности SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - *(по умолчанию)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authPriv”
contextname	string	Имя контекста SNMPv3. Поведение свойства: - *supported* если `version` установлено в ”SNMPv3”

hostprototype.create

Описание

`object hostprototype.create(object/array hostPrototypes)`

Этот метод позволяет создавать новые прототипы узлов сети.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Узлы сети-прототипы для создания.

В дополнение к стандартным свойствам узла сети-прототипа метод принимает следующие параметры.

Parameter	Type	Description
groupLinks	array	Связи групп, которые будут созданы для узла сети-прототипа. Поведение параметра: - *required*
ruleid	ID	ID правила LLD, к которому принадлежит узел сети-прототип. Поведение параметра: - *required*
groupPrototypes	array	Прототипы групп, которые будут созданы для узла сети-прототипа.
macros	object/array	Пользовательские макросы, которые будут созданы для узла сети-прототипа.
tags	object/array	Теги узла сети-прототипа.
interfaces	object/array	Пользовательские интерфейсы узла сети-прототипа.
templates	object/array	Шаблоны, которые будут привязаны к узлу сети-прототипу. Шаблоны должны иметь определенным только свойство `templateid`.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных прототипов узлов сети, указанных в свойстве `hostids`. Порядок возвращаемых ID совпадает с порядком переданных прототипов узлов сети.

Примеры

Создание прототипа узла сети

Создайте прототип узла сети "{#VM.NAME}" по правилу LLD "23542" с прототипом группы "{#HV.NAME}", парой тегов "datacenter": "{#DATACENTER.NAME}" и пользовательским интерфейсом SNMPv2 127.0.0.1:161 с community {$SNMP_COMMUNITY}. Свяжите его с группой узлов сети "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.create",
    "params": {
        "host": "{#VM.NAME}",
        "ruleid": "23542",
        "custom_interfaces": "1",
        "groupLinks": [
            {
                "groupid": "2"
            }
        ],
        "groupPrototypes": [
            {
                "name": "{#HV.NAME}"
            }
        ],
        "tags": [
```

```
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            }
        ],
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

См. также

- Ссылка на группу
- Прототип группы
- Тег прототипа узла сети
- Пользовательский интерфейс
- Пользовательский макрос

Источник

CHostPrototype::create() в *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Описание

`object hostprototype.delete(array hostPrototypeIds)`

Этот метод позволяет удалять прототипы узлов сети.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых прототипов узлов сети

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных прототипов узлов сети, указанных в свойстве `hostids`.

Примеры

Удаление нескольких прототипов узлов сети

Удаление двух прототипов узлов сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Источник

CHostPrototype::delete() в *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Описание

`integer/array hostprototype.get(object parameters)`

Метод позволяет получать прототипы узлов сети в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
hostids	ID/array	Возвращать только прототипы узлов сети с указанными ID.
discoveryids	ID/array	Возвращать только прототипы узлов сети, которые принадлежат указанным правилам LLD.
inherited	boolean	Если установлено в `true`, возвращать только элементы данных, унаследованные от шаблона.

Parameter	Type	Description
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения прототипа узла сети. Объект обнаружения прототипа узла сети связывает обнаруженный прототип узла сети с прототипом узла сети, из которого он был обнаружен. Он имеет следующие свойства: `host` - (`string`) ID узла сети; `parent_hostid` - (`string`) ID прототипа узла сети, из которого был создан прототип узла сети; `status` - (`int`) статус обнаружения прототипа узла сети: 0 - *(default)* прототип узла сети обнаружен, 1 - прототип узла сети больше не обнаружен; `ts_delete` - (`timestamp`) время, когда прототип узла сети, который больше не обнаружен, будет удален; `ts_disable` - (`timestamp`) время, когда прототип узла сети, который больше не обнаружен, будет отключен; `disable_source` - (`int`) признак того, был ли прототип узла сети отключен правилом LLD или вручную: 0 - *(default)* отключен автоматически, 1 - отключен правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, к которому принадлежит прототип узла сети.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, к которому принадлежит прототип узла сети.
selectInterfaces	query	Возвращать свойство `interfaces` с пользовательскими интерфейсами прототипа узла сети.
selectGroupLinks	query	Возвращать свойство `groupLinks` со связями групп прототипа узла сети.
selectGroupPrototypes	query	Возвращать свойство `groupPrototypes` с прототипами групп прототипа узла сети.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые находятся на связанных шаблонах.
selectMacros	query	Возвращать свойство `macros` с макросами прототипа узла сети.
selectParentHost	query	Возвращать свойство `parentHost` с узлом сети, к которому принадлежит прототип узла сети.
selectTags	query	Возвращать свойство `tags` с тегами прототипа узла сети.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, связанными с прототипом узла сети. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;

- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение прототипов узлов сети из правила LLD

Получите все прототипы узлов сети, их связи с группами, прототипы групп и теги из правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "discover": "0",
            "custom_interfaces": "1",
            "uuid": "051a1469d4d045cbbf818fcc843a352e",
            "flags": "2",
            "inventory_mode": "-1",
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "groupLinks": [
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
```

```
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "available": "0",
                    "error": "",
                    "errors_from": "0",
                    "disable_until": "0",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Ссылка на группу
- Прототип группы
- Пользовательский макрос

Источник

CHostPrototype::get() в *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Описание

`object hostprototype.update(object/array hostPrototypes)`

Этот метод позволяет обновлять существующие прототипы узлов сети.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешение на вызов метода может быть отозвано в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипа узла сети, которые нужно обновить.

Свойство `hostid` должно быть определено для каждого прототипа узла сети, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств прототипа узла сети, метод принимает следующие параметры.

Parameter	Type	Description
groupLinks	array	Связи групп, которые заменят текущие связи групп на прототипе узла сети. Поведение параметра: - *только для чтения* для наследуемых объектов

Parameter	Type	Description
groupPrototypes	array	Прототипы групп, которые заменят существующие прототипы групп на прототипе узла сети. Все прототипы групп, не перечисленные в запросе, будут удалены. Поведение параметра: - *только для чтения* для наследуемых объектов
macros	object/array	Пользовательские макросы, которые заменят текущие пользовательские макросы. Все макросы, не перечисленные в запросе, будут удалены.
tags	object/array	Теги прототипа узла сети, которые заменят текущие теги. Все теги, не перечисленные в запросе, будут удалены. Поведение параметра: - *только для чтения* для наследуемых объектов
interfaces	object/array	Пользовательские интерфейсы прототипа узла сети, которые заменят текущие интерфейсы. Объект пользовательского интерфейса должен содержать все свои параметры. Все интерфейсы, не перечисленные в запросе, будут удалены. Поведение параметра: - *поддерживается*, если `custom_interfaces` объекта прототипа узла сети установлено в "use host prototypes custom interfaces" - *только для чтения* для наследуемых объектов
templates	object/array	Шаблоны, которые заменят текущие связанные шаблоны. У шаблонов должно быть определено только свойство `templateid`.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных прототипов узлов сети, указанных в свойстве `hostids`.

Примеры

Отключение прототипа узла сети

Отключите прототип узла сети, то есть установите для его статуса значение "1".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Обновление тегов прототипа узла сети

Замените теги прототипа узла сети новыми.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Обновление пользовательских интерфейсов узла сети-прототипа

Заменяет унаследованные интерфейсы пользовательскими интерфейсами узла сети-прототипа.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "custom_interfaces": "1",
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

См. также

- Ссылка на группу
- Прототип группы
- Тег прототипа узла сети
- Пользовательский интерфейс
- Пользовательский макрос

Источник

CHostPrototype::update() в *ui/include/classes/api/services/CHostPrototype.php*.

Прототип элемента данных

Этот класс предназначен для работы с прототипами элементов данных.

Ссылки на объекты:

- Прототип элемента данных
  - HTTP header
  - HTTP query field
- Тег прототипа элемента данных
- Предобработка прототипа элемента данных

Доступные методы:

- itemprototype.create - создать новые прототипы элементов данных
- itemprototype.delete - удалить прототипы элементов данных
- itemprototype.get - получить прототипы элементов данных
- itemprototype.update - обновить прототипы элементов данных

Объект прототипа элемента данных

Следующие объекты напрямую связаны с API `itemprototype`.

Прототип элемента данных

Объект прототипа элемента данных имеет следующие свойства.

Свойство	Тип	Описание
itemid	ID	ID прототипа элемента данных. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
delay	string	Интервал обновления прототипа элемента данных. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой комбинацию гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы и макросы LLD. При использовании значение должно быть одним макросом. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы можно записывать как два макроса, разделённых косой чертой (например, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Пример: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Поведение свойства: - *обязательно*, если type имеет значение ”Zabbix agent” (0), ”Simple check” (3), ”Zabbix internal” (5), ”External check” (10), ”Database monitor” (11), ”IPMI agent” (12), ”SSH agent” (13), ”TELNET agent” (14), ”Calculated” (15), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”Browser” (22), ”Telemetry query” (24) или если type имеет значение ”Zabbix agent (active)” (7), а key_ не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит прототип элемента данных. Поведение свойства: - *константа* - *обязательно* для операций создания
interfaceid	ID	ID интерфейса узла сети прототипа элемента данных. Поведение свойства: - *обязательно*, если прототип элемента данных принадлежит узлу сети, а type имеет значение ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” или ”SNMP agent” - *поддерживается*, если прототип элемента данных принадлежит узлу сети, а type имеет значение ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”
key_	string	Ключ прототипа элемента данных. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
name	string	Имя прототипа элемента данных. Поддерживает пользовательские макросы. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
type	integer	Тип прототипа элемента данных. Возможные значения: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
url	string	Строка URL. Поддерживает макросы LLD, пользовательские макросы, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
value_type	integer	Тип информации прототипа элемента данных. Возможные значения: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
allow_traps	integer	Разрешить заполнение значения аналогично элементу данных типа trapper. 0 - *(по умолчанию)* не принимать входящие данные; 1 - принимать входящие данные. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent”

Свойство	Тип	Описание
authtype	integer	Метод аутентификации. Возможные значения, если type имеет значение ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если type имеет значение ”HTTP agent”: 0 - *(по умолчанию)* отсутствует; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если type имеет значение ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если type имеет значение ”HTTP agent”)
description	string	Описание прототипа элемента данных.
flags	integer	Источник прототипа элемента данных. Возможные значения: 2 - прототип элемента данных; 6 - обнаруженный прототип элемента данных Поведение свойства: - *только для чтения*
follow_redirects	integer	Следовать перенаправлениям ответа при опросе данных. Возможные значения: 0 - не следовать перенаправлениям; 1 - *(по умолчанию)* следовать перенаправлениям. Поведение свойства: - *поддерживается*, если type имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
headers	array	Массив заголовков, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если type имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
history	string	Единица времени, определяющая срок хранения данных истории. Также поддерживает пользовательский макрос и макрос LLD. По умолчанию: 31d.
http_proxy	string	Строка подключения к прокси HTTP(S). Поведение свойства: - *поддерживается*, если type имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Датчик IPMI. Поведение свойства: - *обязательно*, если type имеет значение ”IPMI agent”, а key_ не имеет значение ”ipmi.get” - *поддерживается*, если type имеет значение ”IPMI agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
jmx_endpoint	string	Пользовательская строка подключения агента JMX. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` имеет значение ”JMX agent”
logtimefmt	string	Формат времени в записях журнала. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”log” - *только для чтения* для унаследованных объектов
master_itemid	ID	ID главного элемента данных. Поведение свойства: - *обязательно*, если `type` имеет значение ”Dependent item” - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* сохранять без изменений; 1 - преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
params	string	Дополнительные параметры в зависимости от типа прототипа элемента данных: - выполняемый скрипт для прототипов элементов данных SSH agent и TELNET agent; - SQL-запрос для прототипов элементов данных Database monitor; - формула для прототипов вычисляемых элементов данных; - скрипт для прототипов элементов данных Script и Browser. Поведение свойства: - *обязательно*, если `type` имеет значение ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”Script” или ”Browser”)
parameters	object/array	Дополнительные параметры, если `type` имеет значение ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должен быть уникальным. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Script” или ”Browser” - *только для чтения* для унаследованных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”JMX agent”, а `username` задан - *поддерживается*, если `type` имеет значение ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)

Свойство	Тип	Описание
post_type	integer	Тип тела данных POST, хранящегося в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* необработанные данные. 2 - данные JSON. 3 - данные XML. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”JSON data” или ”XML data” - *поддерживается*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”Raw data” - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key”
query_fields	array	Массив полей запроса, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
request_method	integer	Тип метода запроса. Возможные значения: 0 - *(по умолчанию)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
retrieve_mode	integer	Какая часть ответа должна сохраняться. Возможные значения, если `request_method` имеет значение ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* тело; 1 - заголовки; 2 - сохранять тело и заголовки. Возможные значения, если `request_method` имеет значение ”HEAD”: 1 - заголовки. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
snmp_oid	string	OID SNMP. Поведение свойства: - *обязательно*, если `type` имеет значение ”SNMP agent” - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль для файла ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
status	integer	Состояние прототипа элемента данных. Возможные значения: 0 - *(по умолчанию)* прототип элемента данных включён; 1 - прототип элемента данных отключён; 3 - прототип элемента данных не поддерживается.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Также поддерживает пользовательские макросы или макросы LLD как часть списка, разделённого запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов
templateid	ID	ID прототипа элемента данных родительского шаблона. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса на опрос данных элемента данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m). Также поддерживает пользовательские макросы и макросы LLD. Допустимый диапазон значений: 1-600s. По умолчанию: ”” - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Zabbix agent” (0), ”Simple check” (3), а `key_` не начинается с ”`vmware.`” и ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20), а `snmp_oid` начинается с ”`walk[`” или ”`get[`”, ”Script” (21), ”Browser” (22) - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
trapper_hosts	string	Разрешённые узлы сети. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Zabbix trapper” или если `type` имеет значение ”HTTP agent”, а `allow_traps` имеет значение ”Allow incoming data”. Если свойство не указано, ему автоматически присваивается пользовательский макрос `{$TRAPPER.ALLOWED_HOSTS}` (значение по умолчанию). По умолчанию `{$TRAPPER.ALLOWED_HOSTS}` имеет значение `127.0.0.1,::1`. Если свойству присвоена пустая строка, входящие подключения от всех узлов сети будут запрещены.
trends	string	Единица времени, определяющая срок хранения данных трендов. Также поддерживает пользовательский макрос и макрос LLD. По умолчанию: 365d. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float” или ”numeric unsigned”
units	string	Единицы измерения значения. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float” или ”numeric unsigned” - *только для чтения* для унаследованных объектов
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent” или ”TELNET agent”, либо если `type` имеет значение ”JMX agent”, а `password` задан - *поддерживается*, если `type` имеет значение ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”)
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов элементов данных с уже существующими. Создаётся автоматически, если не указан. Поведение свойства: - *поддерживается*, если прототип элемента данных принадлежит шаблону
valuemapid	ID	ID связанной карты значений. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float”, ”character” или ”numeric unsigned” - *только для чтения* для унаследованных объектов
verify_host	integer	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
verify_peer	integer	Проверять ли подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных объектов
discover	integer	Состояние обнаружения прототипа элемента данных. Возможные значения: 0 - *(по умолчанию)* новые элементы данных будут обнаруживаться; 1 - новые элементы данных обнаруживаться не будут, а существующие элементы данных будут помечены как потерянные.
query	object	Конфигурация запроса телеметрии. Определяет, какие данные запрашивает элемент данных из источника телеметрии, как он фильтрует данные и как агрегирует результат. Поведение свойства: - *обязательно*, если `type` имеет значение "Telemetry query" - *только для чтения* для унаследованных или обнаруженных объектов
time_shift	string	Насколько далеко в прошлое (до текущего времени) заканчивается временной диапазон, за которым элемент данных выполняет запрос. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы и макросы LLD. Допустимый диапазон значений: 0-86400s. По умолчанию: 15s. Поведение свойства: - *поддерживается*, если `type` имеет значение "Telemetry query"
lookback_limit	string	Насколько далеко в прошлое (до текущего времени) элемент данных выполняет запрос данных, если отсутствует предыдущий запрос, от которого можно продолжить (например, при первом запуске элемента данных), или если предыдущий запрос старше `delay` (например, из-за недоступности сервера Zabbix). Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы и макросы LLD. Допустимый диапазон значений: 1-259200s. По умолчанию: 10m. Поведение свойства: - *поддерживается*, если `type` имеет значение "Telemetry query"

Свойство	Тип	Описание
granularity	string	Длительность фрагмента, на который разбивается полный временной диапазон запроса. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы и макросы LLD. Допустимый диапазон значений: 1-86400s. По умолчанию: 15s. Поведение свойства: - *поддерживается*, если type имеет значение "Telemetry query"

HTTP-заголовок

Объект заголовка имеет следующие свойства:

Property	Type	Description
name	string	Имя HTTP-заголовка. Поведение свойства: - *required*
value	string	Значение заголовка. Поведение свойства: - *required*

Поле HTTP-запроса

Объект поля запроса определяет имя и значение, которые используются для указания параметра URL. Он имеет следующие свойства:

Property	Type	Description
name	string	Имя параметра. Property behavior: - *required*
value	string	Значение параметра. Property behavior: - *required*

Запрос телеметрии

Объект запроса телеметрии имеет следующие свойства.

Property	Type	Description
signal_type	integer	Тип сигнала, который запрашивает элемент данных. Возможные значения: 0 - Трассировки; 1 - Метрики; 2 - Журналы. Поведение свойства: - *обязательное*

Property	Type	Description
metric_point_type	integer	Тип точки метрики, который запрашивает элемент данных. Возможные значения: 0 - Сумма; 1 - Показатель; 2 - Гистограмма; 3 - Экспоненциальная гистограмма. Поведение свойства: - *обязательное*, если для `signal_type` установлено значение "Metrics"
columns	array	Массив столбцов данных телеметрии, которые запрашивает элемент данных. Может быть пустым. Поведение свойства: - *обязательное*
aggregated_columns	array	Массив агрегированных столбцов данных телеметрии, которые запрашивает элемент данных, а также способ вычисления значения на основе записей столбцов. Поведение свойства: - *обязательное*
filter	object	Фильтр запроса телеметрии для записей столбцов в запросе. Поведение свойства: - *обязательное*

Столбцы данных телеметрии

Объект столбцов данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, который запрашивает элемент данных. См. поддерживаемые столбцы данных телеметрии. Для нескольких столбцов с одиночными значениями каждое значение `column` должно быть уникальным. Для нескольких столбцов, содержащих пары «ключ-значение», каждая пара `column` и `attribute_key` должна быть уникальной.
attribute_key	string	Имя ключа, который запрашивает элемент данных, в столбце, содержащем пары «ключ-значение». Для нескольких столбцов, содержащих пары «ключ-значение», каждая пара `column` и `attribute_key` должна быть уникальной. Поведение свойства: - *обязательно*, если `column` задан как столбец, содержащий пары «ключ-значение»

Агрегированные столбцы данных телеметрии

Объект агрегированных столбцов данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, который запрашивает элемент данных. См. поддерживаемые столбцы данных телеметрии. Для нескольких столбцов каждое значение `column` должно быть уникальным. Это свойство не поддерживается, если для `function` установлено значение "Count". Поведение свойства: - *обязательно*, если для `function` установлено значение "Min", "Max", "Average", "Sum" или "Percentile"
function	integer	Функция, применяемая элементом данных к записям столбца. Возможные значения: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Поведение свойства: - *обязательно*
parameters	array	Процентиль, который элемент данных вычисляет по записям столбца. Возможное значение: число от 0 до 100, заданное в виде строки. Поведение свойства: - *обязательно*, если для `function` установлено значение "Percentile"
alias	string	Уникальное имя вычисленного значения, используемое для его идентификации в значении элемента данных. Каждое значение `alias` должно быть уникальным. Поведение свойства: - *обязательно*

Фильтр данных телеметрии

Объект фильтра данных телеметрии имеет следующие свойства.

Property	Type	Description
conditions	array	Условия фильтра данных телеметрии, которым должна соответствовать запись в столбце данных телеметрии. Поведение свойства: - *обязательное*

Property	Type	Description
evaltype	integer	Метод вычисления условий фильтра. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Пользовательское выражение. Поведение свойства: - *обязательное*
eval_formula	string	Сгенерированное выражение, используемое для вычисления условий фильтра. Выражение содержит идентификаторы, ссылающиеся на определенные условия фильтра по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *только для чтения*
formula	string	Определенное пользователем выражение, используемое для вычисления условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, ссылающиеся на определенные условия фильтра по их `formulaid`. Идентификаторы, используемые в выражении, должны в точности соответствовать идентификаторам, определенным в условиях фильтра: ни одно условие не должно оставаться неиспользованным или быть пропущенным. Поведение свойства: - *обязательное*, если для `evaltype` установлено значение "Пользовательское выражение"

Условия фильтра данных телеметрии

Объект условий фильтра данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, к которому применяется условие фильтра. См. поддерживаемые столбцы данных телеметрии. Поведение свойства: - *обязательное*
attribute_key	string	Имя ключа, к которому применяется условие, в столбце, содержащем пары «ключ-значение». Поведение свойства: - *обязательное*, если `column` задан как столбец, содержащий пары «ключ-значение»

Property	Type	Description
operator	integer	Оператор условия фильтра. Возможные значения для столбцов с одиночными значениями: 0 - Равно; 1 - Не равно; 2 - Содержит; 3 - Не содержит. Возможные значения для столбцов, содержащих пары «ключ-значение»: 0 - Равно; 1 - Не равно; 12 - Существует. Поведение свойства: - *обязательное*
value	string	Значение для сравнения. Поведение свойства: - *обязательное*, если `operator` задан как ”Равно”, ”Не равно”, ”Содержит” или ”Не содержит”
formulaid	string	Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. Идентификатор должен быть задан пользователем при изменении условий фильтра, но при последующем запросе условий будет сгенерирован заново. Поведение свойства: - *обязательное*, если `evaltype` объекта Фильтр запроса телеметрии задан как ”Пользовательское выражение”

Тег прототипа элемента данных

Объект тега прототипа элемента данных имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега прототипа элемента данных. Поведение свойства: - *required*
value	string	Значение тега прототипа элемента данных. Поведение свойства: - *read-only*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети. Поведение свойства: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *read-only*.

Предобработка прототипа элемента данных

Объект предобработки прототипа элемента данных имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип параметра предобработки. Возможные значения: 1 - Пользовательский множитель; 2 - Удаление символов справа; 3 - Удаление символов слева; 4 - Удаление символов с обеих сторон; 5 - Регулярное выражение; 6 - Преобразование логического значения в десятичное; 7 - Преобразование восьмеричного числа в десятичное; 8 - Преобразование шестнадцатеричного числа в десятичное; 9 - Простое изменение; 10 - Изменение в секунду; 11 - XML XPath; 12 - JSONPath; 13 - В диапазоне; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка наличия ошибки в JSON; 17 - Проверка наличия ошибки в XML; 18 - Проверка наличия ошибки с помощью регулярного выражения; 19 - Отбрасывать неизменившиеся значения; 20 - Отбрасывать неизменившиеся значения с контролем доступности; 21 - JavaScript; 22 - Шаблон Prometheus; 23 - Преобразование Prometheus в JSON; 24 - Преобразование CSV в JSON; 25 - Замена; 26 - Проверка неподдерживаемого значения; 27 - Преобразование XML в JSON; 28 - Значение SNMP walk; 29 - Преобразование SNMP walk в JSON; 30 - Значение SNMP get. Поведение свойства: - *обязательное*

Свойство	Тип	Описание
params	string	Дополнительные параметры, используемые параметром предобработки. Несколько параметров разделяются символом перевода строки (\n). Если для `type` установлено значение ”Проверка неподдерживаемого значения”, параметры соответствуют синтаксису `<scope>[\n<pattern>]`, где *pattern* — регулярное выражение, а *scope* имеет одно из следующих значений: -1 - соответствует любой ошибке; 0 - проверять, соответствует ли сообщение об ошибке значению *pattern*; 1 - проверять, не соответствует ли сообщение об ошибке значению *pattern*. Поведение свойства: - *обязательное*, если для `type` установлено значение ”Пользовательский множитель” (1), ”Удаление символов справа” (2), ”Удаление символов слева” (3), ”Удаление символов с обеих сторон” (4), ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”В диапазоне” (13), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка наличия ошибки в JSON” (16), ”Проверка наличия ошибки в XML” (17), ”Проверка наличия ошибки с помощью регулярного выражения” (18), ”Отбрасывать неизменившиеся значения с контролем доступности” (20), ”JavaScript” (21), ”Шаблон Prometheus” (22), ”Преобразование Prometheus в JSON” (23), ”Преобразование CSV в JSON” (24), ”Замена” (25), ”Проверка неподдерживаемого значения” (26), ”Значение SNMP walk” (28), ”Преобразование SNMP walk в JSON” (29) или ”Значение SNMP get” (30)
error_handler	integer	Тип действия, используемый в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке устанавливается сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Возможные значения, если для `type` установлено значение ”Проверка неподдерживаемого значения”: 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательное*, если для `type` установлено значение ”Пользовательский множитель” (1), ”Регулярное выражение” (5), ”Преобразование логического значения в десятичное” (6), ”Преобразование восьмеричного числа в десятичное” (7), ”Преобразование шестнадцатеричного числа в десятичное” (8), ”Простое изменение” (9), ”Изменение в секунду” (10), ”XML XPath” (11), ”JSONPath” (12), ”В диапазоне” (13), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка наличия ошибки в JSON” (16), ”Проверка наличия ошибки в XML” (17), ”Проверка наличия ошибки с помощью регулярного выражения” (18), ”Шаблон Prometheus” (22), ”Преобразование Prometheus в JSON” (23), ”Преобразование CSV в JSON” (24), ”Проверка неподдерживаемого значения” (26), ”Преобразование XML в JSON” (27), ”Значение SNMP walk” (28), ”Преобразование SNMP walk в JSON” (29) или ”Значение SNMP get” (30)

Свойство	Тип	Описание
error_handler_params	string	Параметры обработчика ошибок. Поведение свойства: - *обязательное*, если для `error_handler` установлено значение ”Установить пользовательское значение” или ”Установить пользовательское сообщение об ошибке”

Для каждого типа предобработки поддерживаются следующие параметры и обработчики ошибок.

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
1	Пользовательский множитель	множитель^1, 6^			0, 1, 2, 3
2	Удаление символов справа	список символов[2]			
3	Удаление символов слева	список символов[2]			
4	Удаление символов с обеих сторон	список символов[2]			
5	Регулярное выражение	шаблон[3]	результат[2]		0, 1, 2, 3
6	Преобразование логического значения в десятичное				0, 1, 2, 3
7	Преобразование восьмеричного числа в десятичное				0, 1, 2, 3
8	Преобразование шестнадцатеричного числа в десятичное				0, 1, 2, 3
9	Простое изменение				0, 1, 2, 3
10	Изменение в секунду				0, 1, 2, 3
11	XML XPath	путь[4]			0, 1, 2, 3
12	JSONPath	путь[4]			0, 1, 2, 3
13	В диапазоне	минимум^1, 6^	максимум^1, 6^		0, 1, 2, 3
14	Соответствует регулярному выражению	шаблон[3]			0, 1, 2, 3
15	Не соответствует регулярному выражению	шаблон[3]			0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
16	Проверка наличия ошибки в JSON	путь[4]			0, 1, 2, 3
17	Проверка наличия ошибки в XML	путь[4]			0, 1, 2, 3
18	Проверка наличия ошибки с помощью регулярного выражения	шаблон[3]	результат[2]		0, 1, 2, 3
19	Отбрасывать неизменившиеся значения				
20	Отбрасывать неизменившиеся значения с контролем доступности	секунды^5, 6^			
21	JavaScript	скрипт[2]			
22	Шаблон Prometheus	шаблон^6, 7^	`value`, `label`, `function`	результат^8, 9^	0, 1, 2, 3
23	Преобразование Prometheus в JSON	шаблон^6, 7^			0, 1, 2, 3
24	Преобразование CSV в JSON	символ-разделитель[2]	символ[2]	0,1	0, 1, 2, 3
25	Замена	искомая строка[2]	замена[2]		
26	Проверка неподдерживаемого значения	область[1]	шаблон^3, 6^		1, 2, 3
27	Преобразование XML в JSON				0, 1, 2, 3
28	Значение SNMP walk	OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS		0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
29	Преобразование SNMP walk в JSON[10]	Имя поля[2]	Префикс OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS	0, 1, 2, 3
30	Значение SNMP get	Формат: 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS			0, 1, 2, 3

[1] целое число или число с плавающей точкой
 [2] строка
 [3] регулярное выражение
 [4] JSONPath или XML XPath
 [5] положительное целое число (с поддержкой суффиксов времени, например 30s, 1m, 2h, 1d)
 [6] пользовательский макрос, макрос LLD
 [7] шаблон Prometheus, соответствующий синтаксису: `<metric name>{<label name>="<label value>", ...} == <value>`. Каждый компонент шаблона Prometheus (метрика, имя метки, значение метки и значение метрики) может быть пользовательским макросом или макросом LLD.
 [8] результат Prometheus, соответствующий синтаксису: `<label name>` (может быть пользовательским макросом или макросом LLD), если в качестве второго параметра выбран `label`.
 [9] Одна из функций агрегации: `sum`, `min`, `max`, `avg`, `count`, если в качестве второго параметра выбрано `function`.
 [10] Поддерживает несколько записей "Имя поля,Префикс OID,Формат", разделенных символом перевода строки.

itemprototype.create

Описание

`object itemprototype.create(object/array itemPrototypes)`

Этот метод позволяет создавать новые прототипы элементов данных.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототип элемента данных для создания.

Дополнительно к стандартным свойствам прототипа элемента данных метод принимает следующие параметры.

Parameter	Type	Description
ruleid	ID	ID правила LLD, которому принадлежит элемент данных. Поведение параметра: *- required*
preprocessing	array	Параметры предобработки прототипа элемента данных.
tags	array	Теги прототипа элемента данных.

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID созданных прототипов элементов данных, указанных в свойстве `itemids`. Порядок возвращаемых ID совпадает с порядком переданных прототипов элементов данных.

Примеры

Создание прототипа элемента данных

Создайте прототип элемента данных для мониторинга свободного места на диске в обнаруженной файловой системе. Обнаруженные элементы должны быть числовыми элементами данных Zabbix agent, обновляемыми каждые 30 секунд.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Free disk space on {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Создание прототипа элемента данных с предварительной обработкой

Создайте элемент данных, используя изменение за секунду и пользовательский множитель в качестве второго шага.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "                {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Создание прототипа зависимого элемента данных

Создайте прототип зависимого элемента данных для прототипа мастер-элемента с ID 44211. Разрешены только зависимости на том же узле сети (шаблон/правило обнаружения), поэтому мастер-элемент и зависимый элемент должны иметь одинаковые hostid и ruleid.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

Создание прототипа элемента данных HTTP-агента

Создайте прототип элемента данных с URL, используя макрос пользователя, поля запроса и пользовательские заголовки.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "api item prototype example",
        "key_": "api_http_item",
        "value_type": 3,
        "url": "{$URL_PROTOTYPE}",
```

```
        "query_fields": [
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value" "100"
            }
        ],
        "headers": [
            {
                "name": "X-Source",
                "value": "api"
            }
        ],
        "delay": "35"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Создание прототипа элемента данных типа скрипт

Создайте простую сборку данных с помощью прототипа элемента данных типа скрипт.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.itemprototype",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Источник

CItemPrototype::create() в *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

Описание

`object itemprototype.delete(array itemPrototypeIds)`

Этот метод позволяет удалять прототипы элементов данных.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых прототипов элементов данных.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных прототипов элементов данных, указанных в свойстве `prototypeids`.

Примеры

Удаление нескольких прототипов элементов данных

Удалите два прототипа элементов данных.
 Зависимые прототипы элементов данных удаляются автоматически, если удаляется мастер-элемент или прототип элемента данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Источник

CItemPrototype::delete() в *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

Описание

`integer/array itemprototype.get(object parameters)`

Метод позволяет получать прототипы элементов данных в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие желаемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
discoveryids	ID/array	Возвращать только прототипы элементов данных, которые принадлежат указанным правилам LLD.
graphids	ID/array	Возвращать только прототипы элементов данных, которые используются в указанных прототипах графиков.
hostids	ID/array	Возвращать только прототипы элементов данных, которые принадлежат указанным узлам сети.
inherited	boolean	Если задано `true`, возвращать только прототипы элементов данных, унаследованные от шаблона.
itemids	ID/array	Возвращать только прототипы элементов данных с указанными ID.
monitored	boolean	Если задано `true`, возвращать только включенные прототипы элементов данных, которые принадлежат контролируемым узлам сети.
templated	boolean	Если задано `true`, возвращать только прототипы элементов данных, которые принадлежат шаблонам.
templateids	ID/array	Возвращать только прототипы элементов данных, которые принадлежат указанным шаблонам.
triggerids	ID/array	Возвращать только прототипы элементов данных, которые используются в указанных прототипах триггеров.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения прототипа элемента данных. Объект обнаружения прототипа элемента данных связывает обнаруженный прототип элемента данных с прототипом элемента данных, из которого он был обнаружен. Он имеет следующие свойства: `parent_itemid` - `(string)` ID прототипа элемента данных, из которого был создан прототип элемента данных; `key_` - `(string)` ключ прототипа элемента данных; `status` - `(int)` статус обнаружения прототипа элемента данных: 0 - *(по умолчанию)* прототип элемента данных обнаружен, 1 - прототип элемента данных больше не обнаружен; `ts_delete` - `(timestamp)` время, когда прототип элемента данных, который больше не обнаружен, будет удален; `ts_disable` - `(timestamp)` время, когда прототип элемента данных, который больше не обнаружен, будет отключен; `disable_source` - `(int)` признак того, был ли прототип элемента данных отключен правилом LLD или вручную: 0 - *(по умолчанию)* отключен автоматически, 1 - отключен правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом низкоуровневого обнаружения, к которому принадлежит прототип элемента данных.

Parameter	Type	Description
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, к которому принадлежит прототип элемента данных.
selectGraphs	query	Возвращать свойство `graphs` с прототипами графиков, в которых используется прототип элемента данных. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, к которым принадлежит прототип элемента данных.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые находятся на шаблоне/узле сети/связанных шаблонах.
selectTags	query	Возвращать теги прототипа элемента данных в свойстве `tags`.
selectTriggers	query	Возвращать свойство `triggers` с прототипами триггеров, в которых используется прототип элемента данных. Поддерживает `count`.
selectPreprocessing	query	Возвращать свойство `preprocessing` с параметрами предварительной обработки прототипа элемента данных.
selectValueMap	query	Возвращать свойство `valuemap` с картой значений прототипа элемента данных.
filter	object	Возвращать только те результаты, которые точно соответствуют заданному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений, с которыми выполняется сравнение. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит прототип элемента данных.
limitSelects	integer	Ограничивает количество записей, возвращаемых подвыборами. Применяется к следующим подвыборам: `selectGraphs` - результаты будут отсортированы по `name`; `selectTriggers` - результаты будут отсортированы по `description`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение прототипов элементов данных из правила LLD

Получить все прототипы элементов данных для указанного ID правила LLD.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
```

```
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "parameters": []
    },
    {
        "itemid": "10010",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10001",
        "name": "Processor load (1 min average per core)",
        "key_": "system.cpu.load[percpu,avg1]",
        "delay": "1m",
        "history": "1w",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "2",
        "interfaceid": "0",
        "description": "                                CPU     ,                 CPU.",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
```

```
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```

Поиск зависимого элемента данных

Найдите один зависимый элемент данных для указанного ID элемента данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25547",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Seconds",
            "key_": "apache.status.uptime.seconds",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
```

```
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```

Поиск прототипа элемента данных HTTP agent

Найдите прототип элемента данных HTTP agent с методом запроса HEAD для указанного ID узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28257",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10254",
```

```
            "name": "discovered",
            "key_": "item[{#INAME}]",
            "delay": "{#IUPDATE}",
            "history": "90d",
            "trends": "30d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
            "status_codes": "",
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


См. также

- Узел сети
- Прототип графика
- Прототип триггера

Источник

CItemPrototype::get() в *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

Описание

`object itemprototype.update(object/array itemPrototypes)`

Этот метод позволяет обновлять существующие прототипы элементов данных.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Свойства прототипа элемента данных, которые необходимо обновить.

Для каждого прототипа элемента данных должно быть определено свойство `itemid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств прототипа элемента данных, метод принимает следующие параметры.

Parameter	Type	Description
preprocessing	array	Параметры предобработки прототипа элемента данных для замены текущих параметров предобработки. Поведение параметра: - *только для чтения* для наследуемых объектов
tags	array	Теги прототипа элемента данных.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных прототипов элементов данных, указанных в свойстве `itemids`.

Примеры

Изменение интерфейса прототипа элемента данных

Измените интерфейс узла сети, который будет использоваться обнаруженными элементами данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Обновление прототипа зависимого элемента данных

Обновляет прототип зависимого элемента данных новым ID прототипа основного элемента данных. Допускаются только зависимости на одном и том же узле сети (шаблон/правило обнаружения), поэтому основной и зависимый элемент данных должны иметь одинаковые hostid и ruleid.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

Обновление прототипа элемента данных HTTP-агента

Измените поля запроса и удалите все пользовательские заголовки.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Обновление параметров предварительной обработки элемента данных

Обновите прототип элемента данных с правилом предварительной обработки элемента данных “Custom multiplier”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Обновление прототипа элемента данных типа скрипт

Обновите прототип элемента данных типа скрипт, указав другой скрипт, и удалите ненужные параметры, которые использовались предыдущим скриптом.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Источник

CItemPrototype::update() в *ui/include/classes/api/services/CItemPrototype.php*.

Регулярное выражение

Этот класс предназначен для работы с глобальными регулярными выражениями.

Ссылки на объекты:

- Регулярное выражение
- Выражения

Доступные методы:

- regexp.create - создать новые регулярные выражения
- regexp.delete - удалить регулярные выражения
- regexp.get - получить регулярные выражения
- regexp.update - обновить регулярные выражения

Объект регулярного выражения

Следующие объекты напрямую связаны с API `regexp`.

Регулярное выражение

Глобальный объект регулярного выражения имеет следующие свойства.

Property	Type	Description
regexpid	ID	ID регулярного выражения. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления.
name	string	Имя регулярного выражения. Поведение свойства: - *обязательно* для операций создания.
description	text	Описание регулярного выражения.
test_string	string	Тестовая строка.

Выражения

Объект expressions имеет следующие свойства.

Property	Type	Description
expression	string	Регулярное выражение. Поведение свойства: - *required*
expression_type	integer	Тип регулярного выражения. Возможные значения: 0 - Содержит строку; 1 - Содержит любую подстроку из списка; 2 - Не содержит строку; 3 - Соответствует регулярному выражению; 4 - Не соответствует регулярному выражению. Поведение свойства: - *required*
exp_delimiter	string	Разделитель выражения. Значение по умолчанию: `","`. Возможные значения: `","` или `"."`, или `"/"`. Поведение свойства: - *supported* if `expression_type` is set to *Contains any substring from list*.

Property	Type	Description
case_sensitive	integer	Учет регистра. Значение по умолчанию: 0. Возможные значения: 0 - Без учета регистра; 1 - С учетом регистра.

regexp.create

Описание

`object regexp.create(object/array regularExpressions)`

Этот метод позволяет создавать новые глобальные регулярные выражения.

> **Note:**
> Этот метод доступен только для типов пользователей *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Регулярные выражения для создания.

В дополнение к стандартным свойствам, метод принимает следующие параметры.

Parameter	Type	Description
expressions	array	Параметры выражений. Поведение параметра: *- required*

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных регулярных выражений, указанных в свойстве`regexpids`.

Примеры

Создание нового глобального регулярного выражения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "                    ,                                                    ",
        "test_string": "/boot",
        "expressions": [
            {
                "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$"
                "expression_type": "4",
                "case_sensitive": "1"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "9"
        ]
    },
    "id": 1
}
```

Источник

CRegexp::create() в *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Описание

`object regexp.delete(array regexpids)`

Этот метод позволяет удалять глобальные регулярные выражения.

> **Note:**
> Этот метод доступен только для типов пользователей *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых регулярных выражений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удалённых регулярных выражений, указанных в свойстве`regexpids`.

Примеры

Удаление нескольких глобальных регулярных выражений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Источник

CRegexp::delete() в *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Описание

`integer/array regexp.get(object parameters)`

Метод позволяет получать глобальные регулярные выражения в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен только *Super Admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
regexpids	ID/array	Возвращать только регулярные выражения с указанными ID.
selectExpressions	query	Возвращать свойство `expressions`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `regexpid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение глобальных регулярных выражений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```


Источник

CRegexp::get() в *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Описание

`object regexp.update(object/array regularExpressions)`

Этот метод позволяет обновлять существующие глобальные регулярные выражения.

> **Note:**
> Этот метод доступен только для типов пользователей *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object/array) Свойства регулярного выражения, которые необходимо обновить.

Свойство regexpid должно быть определено для каждого объекта, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам, метод принимает следующие параметры.

Parameter	Type	Description
expressions	array	Параметры Expressions.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных регулярных выражений, указанных в свойствеregexpids.

Примеры

Обновление глобального регулярного выражения для обнаружения файловых систем.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": "File systems for discovery",
        "description": "                  ,                                                    ",
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CRegexp::update() в *ui/include/classes/api/services/CRegexp.php*.

Роль

Этот класс предназначен для работы с ролями пользователей.

Ссылки на объекты:

- Role

- Role rules
  - UI element
  - Service
  - Service tag
  - Module
  - Action

Доступные методы:

- role.create - создать новые роли пользователей
- role.delete - удалить роли пользователей
- role.get - получить роли пользователей
- role.update - обновить роли пользователей

Объект роли

Следующие объекты напрямую связаны с API `role`.

Роль

Объект role имеет следующие свойства:

Property	Type	Description
roleid	ID	ID роли. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя роли. Поведение свойства: - *обязательно* для операций создания
type	integer	Тип пользователя. Возможные значения: 1 - *(по умолчанию)* Пользователь; 2 - Администратор; 3 - Суперадминистратор. Поведение свойства: - *обязательно* для операций создания
readonly	integer	Указывает, является ли роль доступной только для чтения. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - Да. Поведение свойства: - *только для чтения*

Правила роли

Объект правил роли имеет следующие свойства:

Свойство	Тип	Описание
ui	array	Массив объектов элементов веб-интерфейса.
ui.default_access	integer	Определяет, включен ли доступ к новым элементам веб-интерфейса. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.

Свойство	Тип	Описание
profile.redirect.enforce	integer	Определяет, применяется ли URL перенаправления на уровне роли после аутентификации и может ли пользователь его переопределить. Возможные значения: 0 - *(по умолчанию)* Отключен; 1 - Включен.
profile.redirect.url	string	URL, используемый для перенаправления пользователя после аутентификации.
services.read.mode	integer	Доступ к сервисам только для чтения. Возможные значения: 0 - Доступ только для чтения к сервисам, указанным в свойстве `services.read.list` или соответствующим свойству `services.read.tag`; 1 - *(по умолчанию)* Доступ только для чтения ко всем сервисам.
services.read.list	array	Массив объектов сервисов. Указанным сервисам, включая дочерние сервисы, будет предоставлен доступ только для чтения для роли пользователя. Доступ только для чтения не переопределяет доступ на чтение и запись к сервисам. Поведение свойства: - *поддерживается*, если для `services.read.mode` установлено значение ”0”.
services.read.tag	object	Массив объектов тегов сервисов. Сервисам, соответствующим тегам, включая дочерние сервисы, будет предоставлен доступ только для чтения для роли пользователя. Доступ только для чтения не переопределяет доступ на чтение и запись к сервисам. Поведение свойства: - *поддерживается*, если для `services.read.mode` установлено значение ”0”.
services.write.mode	integer	Доступ к сервисам на чтение и запись. Возможные значения: 0 - *(по умолчанию)* Доступ на чтение и запись к сервисам, указанным в свойстве `services.write.list` или соответствующим свойству `services.write.tag`; 1 - Доступ на чтение и запись ко всем сервисам.
services.write.list	array	Массив объектов сервисов. Указанным сервисам, включая дочерние сервисы, будет предоставлен доступ на чтение и запись для роли пользователя. Доступ на чтение и запись переопределяет доступ только для чтения к сервисам. Поведение свойства: - *поддерживается*, если для `services.write.mode` установлено значение ”0”.

Свойство	Тип	Описание
services.write.tag	object	Массив объектов тегов сервисов. Сервисам, соответствующим тегам, включая дочерние сервисы, будет предоставлен доступ на чтение и запись для роли пользователя. Доступ на чтение и запись переопределяет доступ только для чтения к сервисам. Поведение свойства: - *поддерживается*, если для `services.write.mode` установлено значение ”0”.
modules	array	Массив объектов модулей.
modules.default_access	integer	Определяет, включен ли доступ к новым модулям. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.
api.access	integer	Определяет, включен ли доступ к API. Возможные значения: 0 - *(по умолчанию)* Отключен; 1 - Включен.
api.mode	integer	Режим обработки методов API, перечисленных в свойстве `api`. Возможные значения: 0 - *(по умолчанию)* Список запрещенных методов; 1 - Список разрешенных методов. Поведение свойства: - *поддерживается*, если для `api.access` установлено значение ”1”.
api	array	Массив методов API. Поддерживаются подстановочные знаки, например: `*` (любой метод), `dashboard.*`, `*.export`. Если методы не указаны, доступ ко всем методам API запрещен, если для `api.mode` установлено значение ”1” (список разрешенных методов), и разрешен, если для `api.mode` установлено значение ”0” (список запрещенных методов). Поведение свойства: - *поддерживается*, если для `api.access` установлено значение ”1”.
actions	array	Массив объектов действий.
actions.default_access	integer	Определяет, включен ли доступ к новым действиям. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.
devices.access	integer	Определяет, включен ли доступ к собственным мобильным устройствам пользователя для использования с мобильным приложением Zabbix. Возможные значения: 0 - (*по умолчанию для пользователей и администраторов*) Отключен; 1 - (*по умолчанию для суперадминистраторов*) Включен. Поведение свойства: - *поддерживается*, если включен параметр `EnableMobileDevices`

Свойство	Тип	Описание
devices.actions	array	Массив объектов действий с устройствами. Поведение свойства: - *поддерживается*, если для `devices.access` установлено значение ”1”
devices.actions.default_access	integer	Определяет, включен ли доступ к новым действиям с устройствами. Возможные значения: 0 - (*по умолчанию для пользователей и администраторов*) Отключен; 1 - (*по умолчанию для суперадминистраторов*) Включен. Поведение свойства: - *поддерживается*, если включен параметр `EnableMobileDevices`

Элемент веб-интерфейса

Объект элемента веб-интерфейса имеет следующие свойства:

Property	Type	Description
name	string	Название элемента веб-интерфейса. Возможные значения, если для type объекта Role установлено значение ”User”, ”Admin” или ”Super admin”: monitoring.dashboard - *Панели мониторинга*; monitoring.problems - *Мониторинг > Проблемы*; monitoring.hosts - *Мониторинг > Узлы сети*; monitoring.latest_data - *Мониторинг > Последние данные*; monitoring.maps - *Мониторинг > Карты*; services.services - *Сервисы > Сервисы*; services.sla_report - *Сервисы > Отчет SLA*; inventory.overview - *Инвентаризация > Обзор*; inventory.hosts - *Инвентаризация > Узлы сети*; reports.availability_report - *Отчеты > Отчет о доступности*; reports.top_triggers - *Отчеты > Топ-100 триггеров*. Возможные значения, если для type объекта Role установлено значение ”Admin” или ”Super admin”: monitoring.discovery - *Мониторинг > Обнаружение*; services.sla - *Сервисы > SLA*; reports.scheduled_reports - *Отчеты > Запланированные отчеты*; reports.notifications - *Отчеты > Уведомления*; configuration.template_groups - *Сбор данных > Группы шаблонов*; configuration.host_groups - *Сбор данных > Группы узлов сети*; configuration.templates - *Сбор данных > Шаблоны*; configuration.hosts - *Сбор данных > Узлы сети*; configuration.maintenance - *Сбор данных > Обслуживание*; configuration.discovery - *Сбор данных > Обнаружение*; configuration.trigger_actions - *Оповещения > Действия > Действия с триггерами*; configuration.service_actions - *Оповещения > Действия > Действия с сервисами*; configuration.discovery_actions - *Оповещения > Действия > Действия обнаружения*; configuration.autoregistration_actions - *Оповещения > Действия > Действия автоматической регистрации*; configuration.internal_actions - *Оповещения > Действия > Внутренние действия*. Возможные значения, если для type объекта Role установлено значение ”Super admin”: reports.system_info - *Отчеты > Сведения о системе*; reports.audit - *Отчеты > Журнал аудита*; reports.action_log - *Отчеты > Журнал действий*; configuration.event_correlation - *Сбор данных > Корреляция событий*; administration.media_types - *Оповещения > Типы носителей*; administration.scripts - *Оповещения > Скрипты*; administration.user_groups - *Пользователи > Группы пользователей*; administration.user_roles - *Пользователи > Роли пользователей*; administration.users - *Пользователи > Пользователи*; administration.api_tokens - *Пользователи > Токены API*; administration.authentication - *Пользователи > Аутентификация*; administration.linked_devices - *Пользователи > Устройства*; administration.general - *Администрирование > Общие параметры*; administration.audit_log - *Администрирование > Журнал аудита*; administration.housekeeping - *Администрирование > Очистка*

Property	Type	Description
status	integer	Определяет, включен ли доступ к элементу веб-интерфейса. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.

Сервис

Property	Type	Description
serviceid	ID	ID службы. Поведение свойства: - *обязательно*

Тег сервиса

Property	Type	Description
tag	string	Имя тега. Если указана пустая строка, тег сервиса не будет использоваться для сопоставления сервиса. Property behavior: - *required*
value	string	Значение тега. Если значение не указано или указана пустая строка, для сопоставления сервиса будет использоваться только имя тега.

Модуль

Объект module имеет следующие свойства:

Property	Type	Description
moduleid	ID	ID модуля. Поведение свойства: - *required*
status	integer	Включен ли доступ к модулю. Возможные значения: 0 - Disabled; 1 - *(default)* Enabled.

Действие

Объект действия имеет следующие свойства:

Property	Type	Description
name	string	Название действия. Возможные значения, если для свойства `type` объекта Role установлено значение ”User”, ”Admin” или ”Super admin”: `edit_dashboards` - Создание и редактирование панелей; `edit_maps` - Создание и редактирование карт; `add_problem_comments` - Добавление комментариев к проблемам; `change_severity` - Изменение важности проблемы; `acknowledge_problems` - Подтверждение проблем; `suppress_problems` - Подавление проблем; `close_problems` - Закрытие проблем; `execute_scripts` - Выполнение сценариев; `manage_api_tokens` - Управление токенами API; `change_problem_ranking` - Изменение ранжирования проблемы с причины на симптом и наоборот; `edit_own_media` - Разрешение пользователям создавать и редактировать собственные способы оповещения. Возможные значения, если для свойства `type` объекта Role установлено значение ”Admin” или ”Super admin”: `edit_maintenance` - Создание и редактирование периодов обслуживания; `manage_scheduled_reports` - Управление запланированными отчетами, `manage_sla` - Управление SLA, `select_server_for_monitoring` - Разрешение пользователям создавать узлы сети, отслеживаемые сервером, а также правилами обнаружения сети, выполняемыми сервером. Возможные значения, если для свойства `type` объекта Role установлено значение ”User” или ”Admin”: `invoke_execute_now` - разрешает пользователям, имеющим только права на чтение узла сети, выполнять проверки элементов данных. Возможные значения, если для свойства `type` объекта Role установлено значение ”Super admin”: `edit_user_media` - Разрешение пользователям создавать и редактировать способы оповещения для других пользователей. Поведение свойства: *- обязательное*
status	integer	Определяет, включен ли доступ для выполнения действия. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.

Действие с устройством

Объект действия с устройством имеет следующие свойства:

Property	Type	Description
name	string	Название действия с устройством. Возможные значения, если для свойства `type` объекта Role установлено значение ”User”, ”Admin” или ”Super admin”: `manage_own` - Разрешить пользователям добавлять или удалять собственные мобильные устройства. Возможные значения, если для свойства `type` объекта Role установлено значение ”Super admin”: `manage_user` - Разрешить пользователям добавлять или удалять мобильные устройства всех пользователей. Поведение свойства: - *required*
status	integer	Определяет, разрешен ли доступ для выполнения действия. Возможные значения: 0 - Отключено; 1 - (*default*) Включено.

role.create

Описание

`object role.create(object/array roles)`

Этот метод позволяет создавать новые роли.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Роли для создания.

Помимо стандартных свойств роли, метод принимает следующие параметры.

Parameter	Type	Description
rules	array	Правила роли, которые будут созданы для роли.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных ролей, указанных в свойстве `roleids`. Порядок возвращаемых ID совпадает с порядком переданных ролей.

Примеры

Создание роли

Создайте роль с типом ”User” и запретом доступа к двум элементам интерфейса.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "role.create",
    "params": {
        "name": "Operator",
        "type": "1",
        "rules": {
            "ui": [
                {
```

```
                    "name": "monitoring.hosts",
                    "status": "0"
                },
                {
                    "name": "monitoring.maps",
                    "status": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

См. также

- Правила роли
- Элемент интерфейса
- Модуль
- Действие

Источник

CRole::create() в *ui/include/classes/api/services/CRole.php*.

role.delete

Описание

`object role.delete(array roleids)`

Этот метод позволяет удалять роли.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(    )` ID удаляемых ролей.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных ролей, указанных в свойстве `roleids`.

Примеры

Удаление нескольких ролей пользователей

Удаление двух ролей пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
```

```
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CRole::delete() в *ui/include/classes/api/services/CRole.php*.

role.get

Описание

`integer/array role.get(object parameters)`

Метод позволяет получать роли в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
roleids	ID/array	Возвращать только роли с указанными идентификаторами.
current	boolean	Если установлено значение `true`, возвращать только роль текущего пользователя.
selectRules	query	Возвращать свойство `rules`, содержащее правила роли.
selectUsers	query	Возвращать свойство `users`, содержащее пользователей, которым назначена роль. Ограничения в зависимости от типа пользователя см. в описании метода `user.get`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `roleid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	

Параметр	Тип	Описание
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение данных роли

Получите данные роли «Роль супер-администратора» и правила её доступа.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "      -        ",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.latest_data",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.maps",
                        "status": "1"
                    },
                    {
```

```
            "name": "services.services",
            "status": "1"
        },
        {
            "name": "services.sla_report",
            "status": "1"
        },
        {
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
```

```
            },
            {
                "name": "configuration.trigger_actions",
                "status": "1"
            },
            {
                "name": "configuration.service_actions",
                "status": "1"
            },
            {
                "name": "configuration.discovery_actions",
                "status": "1"
            },
            {
                "name": "configuration.autoregistration_actions",
                "status": "1"
            },
            {
                "name": "configuration.internal_actions",
                "status": "1"
            },
            {
                "name": "reports.system_info",
                "status": "1"
            },
            {
                "name": "reports.audit",
                "status": "1"
            },
            {
                "name": "reports.action_log",
                "status": "1"
            },
            {
                "name": "configuration.event_correlation",
                "status": "1"
            },
            {
                "name": "administration.media_types",
                "status": "1"
            },
            {
                "name": "administration.scripts",
                "status": "1"
            },
            {
                "name": "administration.user_groups",
                "status": "1"
            },
            {
                "name": "administration.user_roles",
                "status": "1"
            },
            {
                "name": "administration.users",
                "status": "1"
            },
            {
                "name": "administration.api_tokens",
                "status": "1"
            },
            {
```

```
                "name": "administration.authentication",
                "status": "1"
            },
            {
                "name": "administration.general",
                "status": "1"
            },
            {
                "name": "administration.audit_log",
                "status": "1"
            },
            {
                "name": "administration.housekeeping",
                "status": "1"
            },
            {
                "name": "administration.proxy_groups",
                "status": "1"
            },
            {
                "name": "administration.proxies",
                "status": "1"
            },
            {
                "name": "administration.macros",
                "status": "1"
            },
            {
                "name": "administration.queue",
                "status": "1"
            }
        ],
        "ui.default_access": "1",
        "profile.redirect.enforce": "0",
        "profile.redirect.url": "",
        "services.read.mode": "1",
        "services.read.list": [],
        "services.read.tag": {
            "tag": "",
            "value": ""
        },
        "services.write.mode": "1",
        "services.write.list": [],
        "services.write.tag": {
            "tag": "",
            "value": ""
        },
        "modules": [
            {
                "moduleid": 1,
                "status": "1"
            },
            {
                "moduleid": 2,
                "status": "1"
            },
            {
                "moduleid": 3,
                "status": "1"
            },
            {
                "moduleid": 4,
```

```
            "status": "1"
        },
        {
            "moduleid": 5,
            "status": "1"
        },
        {
            "moduleid": 6,
            "status": "1"
        },
        {
            "moduleid": 7,
            "status": "1"
        },
        {
            "moduleid": 8,
            "status": "1"
        },
        {
            "moduleid": 9,
            "status": "1"
        },
        {
            "moduleid": 10,
            "status": "1"
        },
        {
            "moduleid": 11,
            "status": "1"
        },
        {
            "moduleid": 12,
            "status": "1"
        },
        {
            "moduleid": 13,
            "status": "1"
        },
        {
            "moduleid": 14,
            "status": "1"
        },
        {
            "moduleid": 15,
            "status": "1"
        },
        {
            "moduleid": 16,
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
```

```
            {
                "moduleid": 20,
                "status": "1"
            },
            {
                "moduleid": 21,
                "status": "1"
            },
            {
                "moduleid": 22,
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
```

```
                {
                    "name": "edit_dashboards",
                    "status": "1"
                },
                {
                    "name": "edit_maps",
                    "status": "1"
                },
                {
                    "name": "acknowledge_problems",
                    "status": "1"
                },
                {
                    "name": "suppress_problems",
                    "status": "1"
                },
                {
                    "name": "close_problems",
                    "status": "1"
                },
                {
                    "name": "change_severity",
                    "status": "1"
                },
                {
                    "name": "add_problem_comments",
                    "status": "1"
                },
                {
                    "name": "execute_scripts",
                    "status": "1"
                },
                {
                    "name": "manage_api_tokens",
                    "status": "1"
                },
                {
                    "name": "edit_maintenance",
                    "status": "1"
                },
                {
                    "name": "manage_scheduled_reports",
                    "status": "1"
                },
                {
                    "name": "manage_sla",
                    "status": "1"
                },
                {
                    "name": "invoke_execute_now",
                    "status": "1"
                },
                {
                    "name": "change_problem_ranking",
                    "status": "1"
                },
                {
                    "name": "edit_own_media",
                    "status": "1"
                },
                {
                    "name": "edit_user_media",
```

```
                        "status": "1"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```

См. также

- Правила роли
- Пользователь

Источник

CRole::get() в *ui/include/classes/api/services/CRole.php*.

role.update

Описание

`object role.update(object/array roles)`

Этот метод позволяет обновлять существующие роли.

> **Note:**
> Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства роли, которые необходимо обновить.

Свойство `roleid` должно быть определено для каждой роли, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Дополнительно к стандартным свойствам роли метод принимает следующие параметры.

Параметр	Тип	Описание
rules	array	Доступ rules, которые нужно обновить для роли.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых ролей, указанных в свойстве `roleids`.

Примеры

Отключение возможности выполнять скрипты

Обновить роль с ID ”5”, отключив возможность выполнять скрипты.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
```

```
                ]
            }
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Ограничение доступа к API

Обновить роль с ID "5", запретив вызов любых методов "create", "update" или "delete".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CRole::update() в *ui/include/classes/api/services/CRole.php*.

Сервис

Этот класс предназначен для работы с ИТ-инфраструктурой и бизнес-сервисами.

Ссылки на объекты:

- Service
- Status rule
- Service tag
- Service alarm

- Problem tag

Доступные методы:

- service.create - создать новые сервисы
- service.delete - удалить сервисы
- service.get - получить сервисы
- service.update - обновить сервисы

Объект service

Следующие объекты напрямую связаны с API `service`.

Сервис

Объект service имеет следующие свойства.

Property	Type	Description
serviceid	ID	ID сервиса. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
algorithm	integer	Правило расчета статуса. Применимо только если существуют дочерние сервисы. Возможные значения: 0 - установить статус в OK; 1 - наиболее критичный, если у всех дочерних сервисов есть проблемы; 2 - наиболее критичный из дочерних сервисов. Поведение свойства: - *обязательно* для операций создания
name	string	Имя сервиса. Поведение свойства: - *обязательно* для операций создания
sortorder	integer	Позиция сервиса, используемая для сортировки. Возможные значения: 0-999. Поведение свойства: - *обязательно* для операций создания
weight	integer	Вес сервиса. Возможные значения: 0-1000000. По умолчанию: 0.

Property	Type	Description
propagation_rule	integer	Правило распространения статуса. Возможные значения: 0 - *(по умолчанию)* распространять статус сервиса как есть - без каких-либо изменений; 1 - увеличить распространяемый статус на заданное `propagation_value` (на 1-5 уровней серьезности); 2 - уменьшить распространяемый статус на заданное `propagation_value` (на 1-5 уровней серьезности); 3 - игнорировать этот сервис - статус вообще не распространяется на родительский сервис; 4 - установить фиксированный статус сервиса, используя заданное `propagation_value`. Поведение свойства: - *обязательно*, если задан `propagation_value`
propagation_value	integer	Значение распространения статуса. Возможные значения, если `propagation_rule` установлено в "0" или "3": 0 - Не классифицировано. Возможные значения, если `propagation_rule` установлено в "1" или "2": 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. Возможные значения, если `propagation_rule` установлено в "4": -1 - OK; 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. Поведение свойства: - *обязательно*, если задано `propagation_rule`
status	integer	Находится ли сервис в состоянии OK или в состоянии проблемы. Если сервис находится в состоянии проблемы, `status` равен либо серьезности наиболее критичной проблемы, либо наивысшему статусу дочернего сервиса в состоянии проблемы. Если сервис находится в состоянии OK, `status` равен -1. Поведение свойства: - *только для чтения*
description	string	Описание сервиса.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных сервисов с уже существующими. Автоматически генерируется, если не указан.
created_at	integer	Unix timestamp, когда сервис был создан.

Property	Type	Description
readonly	integer	Доступ к сервису. Возможные значения: 0 - чтение и запись; 1 - только чтение. Поведение свойства: - *только для чтения*

Правило статуса

Объект правила статуса имеет следующие свойства.

Property	Type	Description
type	integer	Условие для установки статуса (New status). Возможные значения: 0 - если как минимум (N) дочерних сервисов имеют статус (Status) или выше; 1 - если как минимум (N%) дочерних сервисов имеют статус (Status) или выше; 2 - если менее (N) дочерних сервисов имеют статус (Status) или ниже; 3 - если менее (N%) дочерних сервисов имеют статус (Status) или ниже; 4 - если вес дочерних сервисов со статусом (Status) или выше составляет как минимум (W); 5 - если вес дочерних сервисов со статусом (Status) или выше составляет как минимум (N%); 6 - если вес дочерних сервисов со статусом (Status) или ниже меньше (W); 7 - если вес дочерних сервисов со статусом (Status) или ниже меньше (N%). Где: - N (W) — `limit_value`; - (Status) — `limit_status`; - (New status) — `new_status`. Property behavior: - *required*
limit_value	integer	Предельное значение. Возможные значения: - для N и W: 1-100000; - для N%: 1-100. Property behavior: - *required*

Property	Type	Description
limit_status	integer	Предельный статус. Возможные значения: -1 - OK; 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. Property behavior: - *required*
new_status	integer	Новое значение статуса. Возможные значения: 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа. Property behavior: - *required*

Тег сервиса

Объект тега сервиса имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега сервиса. Поведение свойства: - *required*
value	string	Значение тега сервиса.

Сигнал тревоги службы

Note:
Сигналы тревоги службы нельзя напрямую создавать, обновлять или удалять через Zabbix API.

Объекты сигналов тревоги службы представляют изменение состояния службы. Они имеют следующие свойства.

Property	Type	Description
clock	timestamp	Время, когда произошло изменение состояния службы.
value	integer	Статус службы. См. свойство `status` службы для списка возможных значений.

Тег проблемы

Теги проблемы позволяют связывать сервисы с событиями проблемы. Объект тега проблемы имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега проблемы. Property behavior: - *required*
operator	integer	Оператор условия сопоставления. Возможные значения: 0 - *(по умолчанию)* Равно; 2 - Содержит.
value	string	Значение тега проблемы.

service.create

Описание

`object service.create(object/array services)`

Этот метод позволяет создавать новые сервисы.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` создаваемые services.

В дополнение к стандартным свойствам service, метод принимает следующие параметры.

Parameter	Type	Description
children	array	Дочерние services, которые нужно связать с service. Дочерние services должны иметь определенным только свойство `serviceid`.
parents	array	Родительские services, которые нужно связать с service. Родительские services должны иметь определенным только свойство `serviceid`.
tags	array	Service tags, которые нужно создать для service.
problem_tags	array	Problem tags, которые нужно создать для service.
status_rules	array	Status rules, которые нужно создать для service.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных сервисов в свойстве `serviceids`. Порядок возвращаемых идентификаторов соответствует порядку переданных сервисов.

Примеры

Создание сервиса

Создайте сервис, который будет переведен в состояние проблемы, если хотя бы один дочерний элемент находится в состоянии проблемы.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Источник

CService::create() в *ui/include/classes/api/services/CService.php*.

service.delete

Описание

`object service.delete(array serviceIds)`

Этот метод позволяет удалять сервисы.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых услуг.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных услуг, указанных в свойстве `serviceids`.

Примеры

Удаление нескольких сервисов

Удаление двух сервисов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
```

```
    "id": 1
}
```

Источник

CService::delete() в *ui/include/classes/api/services/CService.php*.

service.get

Описание

`integer/array service.get(object parameters)`

Метод позволяет получать сервисы в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
serviceids	ID/array	Возвращать только услуги с указанными ID.
parentids	ID/array	Возвращать только услуги, связанные с указанными родительскими услугами.
deep_parentids	flag	Возвращать все прямые и косвенные дочерние услуги. Используется вместе с `parentids`.
childids	ID/array	Возвращать только услуги, связанные с указанными дочерними услугами.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
tags	object/array	Возвращать только услуги с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все услуги. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
problem_tags	object/array	Возвращать только услуги с указанными тегами проблемы. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все услуги. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
without_problem_tags	flag	Возвращать только услуги без тегов проблемы.
slaids	ID/array	Возвращать только услуги, связанные с указанным SLA.

Parameter	Type	Description
selectChildren	query	Возвращать свойство `children` с дочерними услугами. Поддерживает `count`.
selectParents	query	Возвращать свойство `parents` с родительскими услугами. Поддерживает `count`.
selectTags	query	Возвращать свойство `tags` с тегами услуги. Поддерживает `count`.
selectProblemEvents	query	Возвращать свойство `problem_events` с массивом объектов событий проблемы. Объект события проблемы имеет следующие свойства: `eventid` - (`ID`) ID события; `severity` - (`string`) Текущая серьезность события; `name` - (`string`) Имя разрешенного события. Поддерживает `count`.
selectProblemTags	query	Возвращать свойство `problem_tags` с тегами проблемы. Поддерживает `count`.
selectStatusRules	query	Возвращать свойство `status_rules` с правилами статуса. Поддерживает `count`.
selectStatusTimeline	object/array	Возвращать свойство `status_timeline`, содержащее изменения состояния услуги за указанные периоды. Формат `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` является начальной датой (включительно; целочисленная временная метка), а `period_to` - конечной датой (исключительно; целочисленная временная метка) для интересующего вас периода. Возвращает массив записей, содержащих свойство `start_value` и массив `alarms` для изменений состояния в указанных периодах. Поддерживает `count`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение всех сервисов

Получите все данные обо всех сервисах и их зависимостях.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
```

```
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


Источник

CService::get() в *ui/include/classes/api/services/CService.php*.

service.update

Описание

`object service.update(object/array services)`

Этот метод позволяет обновлять существующие сервисы.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

(object/array) свойства сервиса, которые нужно обновить.

Свойство serviceid должно быть определено для каждого сервиса, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам сервиса, метод принимает следующие параметры.

Parameter	Type	Description
children	array	Дочерние сервисы, которые заменят текущие дочерние сервисы. У дочерних сервисов должно быть определено только свойство serviceid.
parents	array	Родительские сервисы, которые заменят текущие родительские сервисы. У родительских сервисов должно быть определено только свойство serviceid.
tags	array	Теги сервиса, которые заменят текущие теги сервиса.
problem_tags	array	Теги проблемы, которые заменят текущие текущие теги проблемы.
status_rules	array	Правила статуса, которые заменят текущие правила статуса.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы обновленных служб в свойстве serviceids.

Примеры

Установка родительской связи для сервиса

Сделать сервис с ID ”3” родительским для сервиса с ID ”5”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Добавление запланированного простоя

Добавьте простой для сервиса с ID ”4”, запланированный еженедельно с понедельника 22:00 до вторника 10:00.

Запрос:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

Источник

CService::update() в *ui/include/classes/api/services/CService.php*.

Скрипт

Этот класс предназначен для работы со скриптами.

Ссылки на объекты:

- Script
  - Параметры вебхука
- Debug
  - Запись журнала

Доступные методы:

- script.create - создать новые скрипты
- script.delete - удалить скрипты
- script.execute - запускать скрипты
- script.get - получить скрипты
- script.getscriptsbyevents - получить скрипты для событий
- script.getscriptsbyhosts - получить скрипты для узлов сети
- script.update - обновить скрипты

Объект script

Следующие объекты напрямую связаны с API `script`.

Скрипт

Объект скрипта имеет следующие свойства.

Свойство	Тип	Описание
scriptid	ID	Идентификатор скрипта. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
name	string	Имя скрипта. Поведение свойства: - *обязательно* для операций создания
type	integer	Тип скрипта. Возможные значения, если scope имеет значение ”action operation”: 0 - скрипт; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - вебхук. Возможные значения, если scope имеет значение ”manual host action” или ”manual event action”: 6 - URL. Поведение свойства: - *обязательно* для операций создания
command	string	Команда для выполнения. Поведение свойства: - *обязательно*, если type имеет значение ”Script”, ”IPMI”, ”SSH”, ”TELNET” или ”Webhook”
scope	integer	Область действия скрипта. Возможные значения: 1 - операция действия; 2 - ручное действие над узлом сети; 4 - ручное действие над событием. Поведение свойства: - *обязательно* для операций создания
execute_on	integer	Где выполнять скрипт. Возможные значения: 0 - выполнять на агенте Zabbix; 1 - выполнять на сервере Zabbix. Поддерживается *только* если на сервере Zabbix включено выполнение глобальных скриптов; 2 - *(по умолчанию)* выполнять на сервере Zabbix или прокси. Поведение свойства: - *поддерживается*, если type имеет значение ”Script”
menu_path	string	Папки, разделенные косой чертой, образующие навигацию в виде меню в веб-интерфейсе при нажатии на узел сети или событие. Поведение свойства: - *поддерживается*, если scope имеет значение ”manual host action” или ”manual event action”
authtype	integer	Метод аутентификации, используемый для скриптов типа SSH. Возможные значения: 0 - пароль; 1 - открытый ключ. Поведение свойства: - *поддерживается*, если type имеет значение ”SSH”
username	string	Имя пользователя, используемое для аутентификации. Поведение свойства: - *обязательно*, если type имеет значение ”SSH” или ”TELNET”

Свойство	Тип	Описание
password	string	Пароль, используемый для скриптов SSH с аутентификацией по паролю и для скриптов TELNET. Поведение свойства: - *поддерживается*, если `type` имеет значение ”SSH”, а `authtype` имеет значение ”password”, или если `type` имеет значение ”TELNET”
publickey	string	Имя файла открытого ключа, используемого для скриптов SSH с аутентификацией по открытому ключу. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH”, а `authtype` имеет значение ”public key”
privatekey	string	Имя файла закрытого ключа, используемого для скриптов SSH с аутентификацией по открытому ключу. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH”, а `authtype` имеет значение ”public key”
port	string	Номер порта, используемый для скриптов SSH и TELNET. Поведение свойства: - *поддерживается*, если `type` имеет значение ”SSH” или ”TELNET”
groupid	ID	Идентификатор группы узлов сети, в которой можно выполнять скрипт. Если задано значение ”0”, скрипт будет доступен во всех группах узлов сети. По умолчанию: 0.
usrgrpid	ID	Идентификатор группы пользователей, которой будет разрешено выполнять скрипт. Если задано значение ”0”, скрипт будет доступен всем группам пользователей. По умолчанию: 0. Поведение свойства: - *поддерживается*, если `scope` имеет значение ”manual host action” или ”manual event action”
host_access	integer	Права доступа к узлу сети, необходимые для выполнения скрипта. Возможные значения: 2 - *(по умолчанию)* чтение; 3 - запись. Поведение свойства: - *поддерживается*, если `scope` имеет значение ”manual host action” или ”manual event action”
confirmation	string	Текст всплывающего окна подтверждения. Всплывающее окно появится при попытке выполнить скрипт из веб-интерфейса Zabbix. Поведение свойства: - *поддерживается*, если `scope` имеет значение ”manual host action” или ”manual event action”

Свойство	Тип	Описание
timeout	string	Тайм-аут выполнения скрипта вебхука в секундах. Поддерживаются суффиксы времени (например, 30s, 1m). Возможные значения: 1-60 с. По умолчанию: 30 с. Поведение свойства: - *обязательно*, если type имеет значение ”Webhook”
parameters	array	Массив входных параметров вебхука. Поведение свойства: - *поддерживается*, если type имеет значение ”Webhook”
description	string	Описание скрипта.
url	string	URL, заданный пользователем. Поведение свойства: - *обязательно*, если type имеет значение ”URL”
new_window	integer	Открывать URL в новом окне. Возможные значения: 0 - нет; 1 - *(по умолчанию)* да. Поведение свойства: - *поддерживается*, если type имеет значение ”URL”
manualinput	integer	Указывает, принимает ли скрипт ввод, предоставленный пользователем. Возможные значения: 0 - *(по умолчанию)* отключено; 1 - включено; Поведение свойства: - *поддерживается*, если scope имеет значение ”manual host action” или ”manual event action”
manualinput_prompt	string	Текст запроса ручного ввода. Поведение свойства: - *обязательно*, если manualinput имеет значение ”Enabled”
manualinput_validator	string	Строковое поле, используемое для проверки ввода, предоставленного пользователем. Строка содержит регулярное выражение или набор значений, разделенных запятыми. Поведение свойства: - *обязательно*, если manualinput имеет значение ”Enabled”
manualinput_validator_type	integer	Определяет ожидаемый тип пользовательского ввода. Возможные значения: 0 - *(по умолчанию)* строка. Указывает, что manualinput_validator следует рассматривать как регулярное выражение; 1 - список. Указывает, что manualinput_validator следует рассматривать как список возможных значений, разделенных запятыми. Поведение свойства: - *поддерживается*, если manualinput имеет значение ”Enabled”

Свойство	Тип	Описание
manualinput_default_value	string	Значение по умолчанию для автоматического заполнения пользовательского ввода. Поведение свойства: - *поддерживается*, если `manualinput_validator_type` имеет значение "String"

Параметры вебхука

Параметры, передаваемые в скрипт вебхука при его вызове, имеют следующие свойства.

Property	Type	Description
name	string	Имя параметра. Поведение свойства: - *required*
value	string	Значение параметра. Поддерживает макросы.

Отладка

Отладочная информация о выполненном скрипте вебхука. Объект debug имеет следующие свойства.

Property	Type	Description
logs	array	Массив записей журнала.
ms	string	Длительность выполнения скрипта в миллисекундах.

Запись журнала

Объект записи журнала имеет следующие свойства.

Property	Type	Description
level	integer	Уровень журнала.
ms	string	Время, прошедшее в миллисекундах с момента запуска скрипта до добавления записи журнала.
message	string	Сообщение журнала.

script.create

Описание

`object script.create(object/array scripts)`

Этот метод позволяет создавать новые скрипты.

> **Note:**
> Этот метод доступен только пользователям типа *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(/) Создаваемые скрипты.

Этот метод принимает скрипты со стандартными свойствами скрипта.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных скриптов, указанных в свойстве `scriptids`. Порядок возвращаемых ID совпадает с порядком переданных скриптов.

Примеры

Создание скрипта вебхука

Создайте скрипт вебхука, который отправляет HTTP-запрос во внешнюю службу.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Создание SSH-скрипта

Создайте SSH-скрипт с аутентификацией по публичному ключу, который можно выполнять на узле сети и который имеет контекстное меню.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Создание пользовательского скрипта

Создайте пользовательский скрипт, который будет перезагружать сервер. Скрипт потребует права на запись для узла сети и запросит у пользователя ручной ввод. После успешной отправки ввода скрипт отобразит сообщение с подтверждением во веб-интерфейсе.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
        "manualinput_default_value": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Создание скрипта типа URL

Создайте скрипт типа URL для области узла сети, который остается в том же окне и содержит текст подтверждения.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
```

```
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Создание скрипта типа URL с ручным вводом

Создайте скрипт типа URL для области событий, который открывается в новом окне и поддерживает ручной ввод.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

Источник

CScript::create() в *ui/include/classes/api/services/CScript.php*.

script.delete

Описание

`object script.delete(array scriptIds)`

Этот метод позволяет удалять скрипты.

Note:
Этот метод доступен только для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

( ) ID удаляемых скриптов.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных скриптов, указанных в свойстве `scriptids`.

Примеры

Удаление нескольких скриптов

Удаление двух скриптов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Источник

CScript::delete() в *ui/include/classes/api/services/CScript.php*.

script.execute

Описание

`object script.execute(object parameters)`

Этот метод позволяет запускать скрипт на узле сети или событии. Исключение составляют скрипты типа URL; их нельзя выполнять.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, содержащие ID скрипта для запуска, либо ID узла сети, либо ID события и значение manualinput.

Параметр	Тип	Описание
scriptid	ID	ID скрипта для запуска. Поведение параметра: - *обязательно*
hostid	ID	ID узла сети, на котором нужно запустить скрипт. Поведение параметра: - *обязательно*, если `eventid` не задан
eventid	ID	ID события, на котором нужно запустить скрипт. Поведение параметра: - *обязательно*, если `hostid` не задан
manualinput	string	Значение, предоставленное пользователем, с которым нужно запустить скрипт, подставляя макрос {MANUALINPUT}.

Возвращаемые значения

`(object)` Возвращает результат выполнения скрипта.

Property	Type	Description
response	string	Указывает, был ли скрипт выполнен успешно. Возможное значение - `success`.
value	string	Вывод скрипта.
debug	object	Содержит объект debug, если выполняется скрипт вебхука. Для других типов скриптов содержит пустой объект.

Примеры

Выполнить вебхук-скрипт

Выполните вебхук-скрипт, который отправляет HTTP-запрос во внешнюю службу.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
        "debug": {
            "logs": [
                {
                    "level": 3,
```

```
                    "ms": 480,
                    "message": "[Webhook Script] HTTP status: 200."
                }
            ],
            "ms": 495
        }
    },
    "id": 1
}
```


Выполнение пользовательского скрипта

Запустите скрипт "ping" на узле сети.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt
        "debug": []
    },
    "id": 1
}
```


Запуск пользовательского скрипта с ручным вводом

Запустите скрипт "ping" с командой "ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac" на узле сети.

Request:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```

Источник

CScript::execute() в *ui/include/classes/api/services/CScript.php*.

script.get

Описание

`integer/array script.get(object parameters)`

Метод позволяет получать скрипты в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
groupids	ID/array	Возвращать только скрипты, которые можно запускать на указанных группах узлов сети.
hostids	ID/array	Возвращать только скрипты, которые можно запускать на указанных узлах сети.
scriptids	ID/array	Возвращать только скрипты с указанными ID.
usrgrpids	ID/array	Возвращать только скрипты, которые могут запускать пользователи в указанных группах пользователей.
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, на которых можно запускать скрипт.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, на которых можно запускать скрипт.
selectActions	query	Возвращать свойство `actions` с действиями, с которыми связан скрипт.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `scriptid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получить все скрипты

Получить все настроенные скрипты.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
```

```
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
```

```
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
                    "name": "host",
                    "value": "{HOST.HOST}"
                },
                {
                    "name": "v",
                    "value": "2.2"
                }
            ]
        },
        {
            "scriptid": "5",
            "name": "URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "       {HOST.NAME}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
            "new_window": "0",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "6",
            "name": "URL with user input",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "           zabbix {MANUALINPUT}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
```

```
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
            "new_window": "0",
            "manualinput": "1",
            "manualinput_prompt": "                    :",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```


См. также

- Узел сети
- Группа узлов сети

Источник

CScript::get() в *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Описание

`object script.getscriptsbyevents(object parameters)`

Этот метод позволяет получить все доступные скрипты для указанного события или конкретный скрипт, если указан ID скрипта. Если указан manualinput, он подставляет макрос {MANUALINPUT} указанным значением.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Метод принимает объект или массив объектов со следующими параметрами.

Parameter	Type	Description
eventid	ID	ID события, для которого нужно вернуть скрипты. Должен быть уникальным. Поведение параметра: *- required*
scriptid	ID	ID скрипта, который нужно вернуть.
manualinput	string	Значение макроса {MANUALINPUT}, предоставленного пользователем.

Возвращаемые значения

`(object)` Возвращает объект, в котором идентификаторы событий являются свойствами, а массивы доступных скриптов — значениями. Если указан идентификатор скрипта, соответствующее значение представляет собой массив, содержащий конкретный скрипт.

> **Note:**
> Метод автоматически развернет макросы в тексте `confirmation`, тексте `manualinput prompt` и `url`.
> Если указан параметр manualinput, макрос {MANUALINPUT} будет подставлен в указанное значение.

Примеры

Получение скриптов по ID событий

Получить все скрипты, доступные для событий "632" и "614".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
```

```
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "      ,                    *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "     Zabbix        :",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
```

```
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "614": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "1",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
```

```
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "      ,                  *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "     Zabbix        :",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
```

```
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

Получение конкретного скрипта со значением manualinput.

Получить скрипт с ID ”4” для события ”632” со значением manualinput ”dashboard.view”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "      ,                  dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "     Zabbix        :",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
```

```
            "manualinput_default_value": "",
            "parameters": []
          }
        ]
    },
    "id": 1
}
```

Источник

CScript::getScriptsByEvents() в *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Описание

`object script.getscriptsbyhosts(object parameters)`

Этот метод позволяет получить все доступные скрипты на указанном узле сети или конкретный скрипт, если указан ID скрипта. Если указан manualinput, он подставляет макрос `{MANUALINPUT}` со значением, заданным пользователем.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Метод принимает объект или массив объектов со следующими параметрами.

Parameter	Type	Description
hostid	ID	ID узла сети, для которого нужно вернуть скрипты. Должен быть уникальным. Поведение параметра: - *required*
scriptid	ID	ID скрипта, который нужно вернуть.
manualinput	string	Значение макроса {MANUALINPUT}, заданного пользователем.

Возвращаемые значения

`(object)` Возвращает объект, в котором идентификаторы узлов сети являются свойствами, а массивы доступных скриптов — значениями. Если указан идентификатор скрипта, соответствующее значение представляет собой массив, содержащий конкретный скрипт.

> **Note:**
> Метод автоматически развернет макросы в тексте `confirmation`, тексте `manualinput prompt` и `url`.
> Если указан параметр manualinput, макрос {MANUALINPUT} будет подставлен в указанное значение.

Примеры

Получение скриптов по ID узлов сети

Получить все скрипты, доступные на узлах сети ”30079” и ”30073”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
```

```
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
```

```
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
```

```
            "parameters": []
        }
    ],
    "30073": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
```

```
            {
                "scriptid": "4",
                "name": "Open Zabbix page",
                "command": "",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "Are you sure you want to open page *UNKNOWN*?",
                "type": "6",
                "execute_on": "2",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
                "new_window": "1",
                "manualinput": "1",
                "manualinput_prompt": "Zabbix page to open:",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "2",
                "name": "Traceroute",
                "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

Получение конкретного скрипта со значением manualinput.

Получить скрипт с ID "4" на узле сети "30079" со значением manualinput "dashboard.view".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
      "30079": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "      ,                  dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "     Zabbix        :",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

Источник

CScript::getScriptsByHosts() в *ui/include/classes/api/services/CScript.php*.

script.update

Описание

`object script.update(object/array scripts)`

Этот метод позволяет обновлять существующие скрипты.

> **Note:**
> Этот метод доступен только пользователю типа *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства скрипта, которые нужно обновить.

Свойство `scriptid` должно быть определено для каждого скрипта, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений. Исключение составляет изменение свойства `type` с 5 (Webhook) на другое: свойство `parameters` будет очищено.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных скриптов в свойстве `scriptids`.

Примеры

Изменение команды скрипта

Измените команду скрипта на "/bin/ping -c 10 {HOST.CONN} 2>&1".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Изменение команды скрипта и добавление ручного ввода

Измените команду скрипта на "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "           ICMP-    ,                          ping",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Источник

CScript::update() в *ui/include/classes/api/services/CScript.php*.

Событие

Этот класс предназначен для работы с событиями.

Ссылки на объекты:

- Событие
- Тег события
- URL типа медиа

Доступные методы:

- event.get - получить события
- event.acknowledge - подтвердить события

Объект события

Следующие объекты напрямую связаны с API `event`.

Событие

> **Note:**
> События создаются сервером Zabbix и не могут быть изменены через API.

Объект события имеет следующие свойства.

Property	Type	Description
eventid	ID	ID события.
source	integer	Тип события. Возможные значения: 0 - событие, созданное триггером; 1 - событие, созданное правилом обнаружения; 2 - событие, созданное автоматической регистрацией активного агента; 3 - внутреннее событие; 4 - событие, созданное при обновлении статуса сервиса.

Property	Type	Description
object	integer	Тип объекта, связанного с событием. Возможные значения, если `source` имеет значение ”event created by a trigger”: 0 - триггер. Возможные значения, если `source` имеет значение ”event created by a discovery rule”: 1 - обнаруженный узел сети; 2 - обнаруженный сервис. Возможные значения, если `source` имеет значение ”event created by active agent autoregistration”: 3 - автоматически зарегистрированный узел сети. Возможные значения, если `source` имеет значение ”internal event”: 0 - триггер; 4 - элемент данных; 5 - правило LLD. Возможные значения, если `source` имеет значение ”event created on service status update”: 6 - сервис.
objectid	ID	ID связанного объекта.
acknowledged	integer	Было ли событие подтверждено.
clock	timestamp	Время создания события.
ns	integer	Наносекунды в момент создания события.
name	string	Разрешенное имя события.
value	integer	Состояние связанного объекта. Возможные значения, если `source` имеет значение ”event created by a trigger” или ”event created on service status update”: 0 - OK; 1 - проблема. Возможные значения, если `source` имеет значение ”event created by a discovery rule”: 0 - узел сети или сервис доступен; 1 - узел сети или сервис недоступен; 2 - узел сети или сервис обнаружен; 3 - узел сети или сервис потерян. Возможные значения, если `source` имеет значение ”internal event”: 0 - состояние ”normal”; 1 - состояние ”unknown” или ”not supported”. Поведение свойства: - *supported*, если `source` имеет значение ”event created by a trigger”, ”event created by a discovery rule”, ”internal event” или ”event created on service status update”
severity	integer	Текущая серьезность события. Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.

Property	Type	Description
r_eventid	ID	ID события восстановления.
c_eventid	ID	ID события, которое было использовано для переопределения (закрытия) текущего события по правилу глобальной корреляции. См. `correlationid`, чтобы определить точное правило корреляции. Этот параметр определяется только тогда, когда событие закрыто правилом глобальной корреляции.
cause_eventid	ID	ID события-причины.
correlationid	ID	ID правила корреляции, которое инициировало закрытие проблемы. Этот параметр определяется только тогда, когда событие закрыто правилом глобальной корреляции.
userid	ID	ID пользователя, который закрыл событие (если событие было закрыто вручную).
suppressed	integer	Подавлено ли событие. Возможные значения: 0 - событие находится в нормальном состоянии; 1 - событие подавлено.
opdata	string	Операционные данные с раскрытыми макросами.
urls	array	Активные URL-адреса типа media type.

Тег события

Объект тега события имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега события.
value	string	Значение тега события.

URL типа оповещения

Объект URL типа оповещения имеет следующие свойства.

Property	Type	Description
name	string	Имя URL, определенное для типа оповещения.
url	string	Значение URL, определенное для типа оповещения.

Результаты будут содержать записи только для активных типов оповещения с включенным пунктом меню событий. Макрос, используемый в свойствах, будет раскрыт, но если одно из свойств содержит нераскрытый макрос, оба свойства будут исключены из результатов. Поддерживаемые макросы см. в разделе *Поддерживаемые макросы*.

event.acknowledge

Описание

`object event.acknowledge(object/array parameters)`

Этот метод позволяет обновлять события. Можно выполнить следующие действия по обновлению:

- Закрыть событие. Если событие уже разрешено, это действие будет пропущено.
- Подтвердить событие. Если событие уже подтверждено, это действие будет пропущено.
- Снять подтверждение с события. Если событие не подтверждено, это действие будет пропущено.
- Добавить сообщение.
- Изменить серьезность события. Если у события уже такая же серьезность, это действие будет пропущено.
- Подавить событие. Если событие уже подавлено, это действие будет пропущено.
- Снять подавление с события. Если событие не подавлено, это действие будет пропущено.
- Изменить ранг события на cause. Если событие уже имеет ранг cause, это действие будет пропущено.
- Изменить ранг события на symptom. Если событие уже имеет ранг symptom, это действие будет пропущено.

Attention:
Можно обновлять только события триггера.
 Можно обновлять только проблемные события.
 Для закрытия события или изменения его серьезности требуются права чтения/записи для триггера.
 Чтобы закрыть событие, в триггере должно быть разрешено ручное закрытие.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Параметры, содержащие идентификаторы событий и операции обновления, которые необходимо выполнить.

Parameter	Type	Description
eventids	ID/array	Идентификаторы событий, которые необходимо подтвердить. Поведение параметра: - *обязательный*
action	integer	Действие или действия по обновлению события. Возможные значения битовой маски: 1 - закрыть проблему; 2 - подтвердить событие; 4 - добавить сообщение; 8 - изменить важность; 16 - снять подтверждение события; 32 - подавить событие; 64 - отменить подавление события; 128 - изменить ранг события на причину; 256 - изменить ранг события на симптом. Это поле является битовой маской; допускается любая сумма возможных значений битовой маски (например, 6 для подтверждения события и добавления сообщения или 34 для подтверждения и подавления события). Поведение параметра: - *обязательный*
cause_eventid	ID	Идентификатор события-причины. Поведение параметра: - *обязательный*, если `action` содержит бит «изменить ранг события на симптом»
message	string	Текст сообщения. Поведение параметра: - *обязательный*, если `action` содержит бит «добавить сообщение»
severity	integer	Новая важность событий. Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа. Поведение параметра: - *обязательный*, если `action` содержит бит «изменить важность»

Parameter	Type	Description
suppress_until	integer	Метка времени Unix, до которой событие должно быть подавлено. Если задано значение «0», подавление будет бессрочным. Поведение параметра: - *обязательный*, если `action` содержит бит «подавить событие»

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных событий в свойстве `eventids`.

Примеры

Подтверждение события

Подтвердите одно событие и оставьте сообщение.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "         ."
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Изменение серьезности события

Измените серьезность для нескольких событий и оставьте сообщение.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
```

```
            "20428"
        ]
    },
    "id": 1
}
```

Подавление события на неопределенный срок

Подавить одно событие без времени истечения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Снятие подавления события

Снимите подавление с одного ранее подавленного события.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Подтвердить и подавить событие

Подтвердите одно или несколько событий, добавьте сообщение и подавите их в одном запросе.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Suppressing during planned maintenance.",
        "suppress_until": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427,
            20428
        ]
    },
    "id": 1
}
```

Источник

CEvent::acknowledge() в *ui/include/classes/api/services/CEvent.php*.

event.get

Описание

`integer/array event.get(object parameters)`

Метод позволяет получать события в соответствии с заданными параметрами.

Attention:
Этот метод может возвращать события удаленной сущности, если эти события еще не были удалены housekeeper.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
eventids	ID/array	Возвращать только события с указанными ID.
groupids	ID/array	Возвращать только события, созданные объектами, которые принадлежат указанным группам узлов сети.
hostids	ID/array	Возвращать только события, созданные объектами, которые принадлежат указанным узлам сети.
objectids	ID/array	Возвращать только события, созданные указанными объектами.

Parameter	Type	Description
source	integer	Возвращать только события с указанным типом. Список поддерживаемых типов событий см. на странице объекта события. По умолчанию: 0 - события триггера.
object	integer	Возвращать только события, созданные объектами указанного типа. Список поддерживаемых типов объектов см. на странице объекта события. По умолчанию: 0 - триггер.
acknowledged	boolean	Если задано `true`, возвращать только подтвержденные события.
action	integer	Возвращать только события, для которых были выполнены указанные действия обновления события. Для нескольких действий используйте сумму любых допустимых значений битовой карты в качестве битовой маски (например, 34 для подтверждения и подавления события).
action_userids	ID/array	Возвращать только события с указанными ID пользователей, выполнивших действия обновления события.
suppressed	boolean	Если задано `true`, возвращать только подавленные события.
symptom	boolean	Если задано `true`, возвращать только события-симптомы.
severities	integer/array	Возвращать только события с указанными уровнями серьезности события. Применяется только если object - trigger.
trigger_severities	integer/array	Возвращать только события с указанными уровнями серьезности триггера. Применяется только если object - trigger.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И; 2 - ИЛИ.
tags	array	Возвращать только события с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все события. Возможные значения operator: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
eventid_from	string	Возвращать только события с ID больше или равным указанному ID.
eventid_till	string	Возвращать только события с ID меньше или равным указанному ID.
time_from	timestamp	Возвращать только события, созданные в указанное время или позже.
time_till	timestamp	Возвращать только события, созданные в указанное время или раньше.
problem_time_from	timestamp	Возвращает только события, которые находились в состоянии проблемы начиная с `problem_time_from`, независимо от их текущего статуса. Применяется только когда source - событие триггера, а object - trigger. Этот параметр обязателен, если указан `problem_time_till`.
problem_time_till	timestamp	Возвращает только события, которые находились в состоянии проблемы до `problem_time_till`, независимо от их текущего статуса. Применяется только когда source - событие триггера, а object - trigger. Этот параметр обязателен, если указан `problem_time_from`.

Parameter	Type	Description
value	integer/array	Возвращать только события с указанными значениями.
selectAcknowledges	query	Возвращать свойство `acknowledges` с обновлениями события. Обновления события сортируются в обратном хронологическом порядке. Объект обновления события имеет следующие свойства: `acknowledgeid` - (`ID`) ID подтверждения; `userid` - (`ID`) ID пользователя, который обновил событие; `clock` - (`timestamp`) время обновления события; `message` - (`string`) текст сообщения; `action` - (`integer`) выполненное действие обновления, см. `event.acknowledge`; `old_severity` - (`integer`) серьезность события до этого действия обновления; `new_severity` - (`integer`) серьезность события после этого действия обновления; `suppress_until` - (`timestamp`) время, до которого событие будет подавлено; `taskid` - (`ID`) ID задачи, если текущее событие находится в процессе изменения ранга; `maintenanceid` - (`ID`) ID обслуживания; `username` - (`string`) имя пользователя, который обновил событие; `name` - (`string`) имя пользователя, который обновил событие; `surname` - (`string`) фамилия пользователя, который обновил событие. Поддерживает `count`.
selectAlerts	query	Возвращать свойство `alerts` с оповещениями, сгенерированными событием. Оповещения сортируются в обратном хронологическом порядке.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, содержащими объект, который создал событие. Поддерживается только для событий, сгенерированных триггерами, элементами данных или правилами LLD.
selectRelatedObject	query	Возвращать свойство `relatedObject` с объектом, который создал событие. Тип возвращаемого объекта зависит от типа события.
selectSuppressionData	query	Возвращать свойство `suppression_data` со списком активных обслуживаний и ручных подавлений: `maintenanceid` - (`ID`) ID обслуживания; `userid` - (`ID`) ID пользователя, который подавил событие; `suppress_until` - (`integer`) время, до которого событие подавлено.
selectTags	query	Возвращать свойство `tags` с тегами события.
filter	object	Возвращать только те результаты, которые в точности соответствуют указанному фильтру. Принимает объект, где ключи - это имена свойств, а значения - это либо одно значение, либо массив значений, с которыми выполняется сравнение. Не поддерживает свойства с типом данных `text`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `eventid`, `objectid`, `clock`. Возможные значения при использовании вместе с `groupBy`: `objectid`. Возможные значения при использовании вместе с `countOutput` и `groupBy`: `objectid`, `rowscount`.

Parameter	Type	Description
groupBy	string/array	Группировать результаты по указанным свойствам. Указанные свойства будут возвращены в результатах. Возможные значения: `objectid`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов
- количество полученных объектов, если использовался параметр `countOutput`, но не использовался параметр `groupBy`
- массив объектов с результатами агрегации, если использовался параметр `groupBy`

Примеры

Получение событий триггера

Получите последние события от триггера ”22395”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "                    (        CPU     1.5        5m)",
            "severity": "0",
            "r_eventid": "0",
```

```
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "          (1m 5m 15m): (1.6328 3.0522 2.2515),     CPU: 2",
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "                    (       CPU    1.5       5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "        .       ,                          .",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
                    "username": "Admin",
```

```
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "          (1m 5m 15m): (1.6328 3.0522 2.2515),     CPU: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Получение событий за период времени

Получите все события, созданные между 17 и 18 октября 2012 года, в обратном хронологическом порядке.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Less than 25% free in the history cache",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

Получение событий, подтвержденных указанным пользователем

Получение событий, подтвержденных пользователем с ID=10

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Получение 5 триггеров с наибольшим числом событий проблемы

Получите 5 триггеров с уровнями серьезности ”Warning”, ”Average”, ”High” или ”Disaster” вместе с количеством событий проблемы за указанный период времени.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "countOutput": true,
```

```
        "groupBy": "objectid",
        "source": 0,
        "object": 0,
        "value": 1,
        "time_from": 1672531200,
        "time_till": 1677628800,
        "trigger_severities": [2, 3, 4, 5],
        "sortfield": ["rowscount"],
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "objectid": "232124",
            "rowscount": "27"
        },
        {
            "objectid": "29055",
            "rowscount": "23"
        },
        {
            "objectid": "253731",
            "rowscount": "18"
        },
        {
            "objectid": "254062",
            "rowscount": "11"
        },
        {
            "objectid": "23216",
            "rowscount": "7"
        }
    ],
    "id": 1
}
```

См. также

- Alert
- Элемент данных
- Узел сети
- Правило LLD
- Сервис
- Триггер

Источник

CEvent::get() в *ui/include/classes/api/services/CEvent.php*.

Тип носителя

Этот класс предназначен для работы с типами носителей.

Ссылки на объекты:

- Тип носителя
  - Параметры вебхука
  - Параметры скрипта
- Шаблон сообщения

Доступные методы:

- mediatype.create - создать новые типы носителей
- mediatype.delete - удалить типы носителей
- mediatype.get - получить типы носителей
- mediatype.update - обновить типы носителей

Объект способа оповещения

Следующие объекты непосредственно связаны с API `mediatype`.

Тип носителя

Объект типа носителя имеет следующие свойства.

Свойство	Тип	Описание
mediatypeid	ID	Идентификатор типа носителя. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Название типа носителя. Поведение свойства: - *обязательно* для операций создания
type	integer	Транспорт, используемый типом носителя. Возможные значения: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook; 5 - Push.
exec_path	string	Имя файла скрипта (например, notification.sh), расположенного в каталоге, указанном в параметре конфигурации сервера AlertScriptsPath. Поведение свойства: - *обязательно*, если для `type` установлено значение ”Script”
gsm_modem	string	Имя последовательного устройства GSM-модема. Поведение свойства: - *обязательно*, если для `type` установлено значение ”SMS”
passwd	string	Пароль аутентификации. Поведение свойства: - *поддерживается*, если для `smtp_authentication` установлено значение ”Normal password”
provider	integer	Поставщик электронной почты. Возможные значения: 0 - *(по умолчанию)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.

Свойство	Тип	Описание
smtp_email	string	Адрес электронной почты, с которого будут отправляться уведомления. Поведение свойства: - *обязательно*, если для `type` установлено значение ”Email”
smtp_helo	string	SMTP HELO. Поведение свойства: - *поддерживается*, если для `type` установлено значение ”Email”
smtp_server	string	SMTP-сервер. Поведение свойства: - *обязательно*, если для `type` установлено значение ”Email”
smtp_port	integer	Порт SMTP-сервера для подключения. По умолчанию: 25. Поведение свойства: - *поддерживается*, если для `type` установлено значение ”Email”
smtp_security	integer	Уровень безопасности SMTP-соединения. Возможные значения: 0 - *(по умолчанию)* None; 1 - STARTTLS; 2 - SSL/TLS. Поведение свойства: - *поддерживается*, если для `type` установлено значение ”Email”
smtp_verify_host	integer	Проверка узла сети SSL для SMTP. Возможные значения: 0 - *(по умолчанию)* No; 1 - Yes. Поведение свойства: - *поддерживается*, если для `smtp_security` установлено значение ”STARTTLS” или ”SSL/TLS”
smtp_verify_peer	integer	Проверка узла сети SSL для SMTP. Возможные значения: 0 - *(по умолчанию)* No; 1 - Yes. Поведение свойства: - *поддерживается*, если для `smtp_security` установлено значение ”STARTTLS” или ”SSL/TLS”
smtp_authentication	integer	Метод аутентификации SMTP. Возможные значения: 0 - *(по умолчанию)* None; 1 - Normal password; 2 - OAuth token. Аутентификация OAuth не разрешена для поставщика электронной почты *Office365 relay*. Поведение свойства: - *поддерживается*, если для `type` установлено значение ”Email”

Свойство	Тип	Описание
redirection_url	string	URL веб-интерфейса Zabbix для перенаправления обратно после авторизации OAuth. По умолчанию: Значение свойства настроек API `url` с частью `zabbix.php?action=oauth.authorize` Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
client_id	string	Идентификатор клиента, зарегистрированный на сервере авторизации OAuth. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
client_secret	string	Секрет клиента, зарегистрированный на сервере авторизации OAuth. Доступен только пользователю типа Super Admin. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
authorization_url	string	URL OAuth с параметрами для получения токенов доступа и обновления. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
token_url	string	URL OAuth для обмена токена авторизации на токены доступа и обновления. Этот URL также используется сервером для обновления недействительного токена доступа. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
tokens_status	integer	Битовая маска состояния токенов. Возможные значения: 0 - (по умолчанию) Оба токена содержат недействительные значения 1 - Токен доступа содержит действительное значение 2 - Токен обновления содержит действительное значение 3 - Оба токена содержат действительные значения. Поведение свойства: - *поддерживается*, если для `smtp_authentication` установлено значение "OAuth token"
access_token	string	Значение токена доступа OAuth. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
access_token_updated	timestamp	Время последнего изменения `access_token`, выполненного сервером при обновлении с помощью `refresh_token` или API при изменении токенов. Поведение свойства: - *поддерживается*, если для `smtp_authentication` установлено значение "OAuth token"

Свойство	Тип	Описание
access_expires_in	integer	Время в секундах, через которое `access_token` устареет и потребуется выполнить запрос к `refresh_url`. Устанавливается сервером Zabbix при обновлении `access_token` или API при изменении токенов. Временная метка вычисляется путем добавления значения `access_token_updated`. Поведение свойства: - *поддерживается*, если для `smtp_authentication` установлено значение "OAuth token"
refresh_token	string	Значение токена обновления OAuth. Поведение свойства: - *обязательно*, если для `smtp_authentication` установлено значение "OAuth token"
status	integer	Определяет, включен ли тип носителя. Возможные значения: 0 - *(по умолчанию)* Включен; 1 - Отключен.
username	string	Имя пользователя. Поведение свойства: - *поддерживается*, если для `smtp_authentication` установлено значение "Normal password"
maxsessions	integer	Максимальное количество оповещений, которые могут обрабатываться параллельно. Возможные значения, если для `type` установлено значение "SMS": 1. Возможные значения, если для `type` установлено значение "Email", "Script" или "Webhook": 0-100. По умолчанию: 1.
maxattempts	integer	Максимальное количество попыток отправки оповещения. Возможные значения: 1-100. По умолчанию: 3.
attempt_interval	string	Интервал между повторными попытками. Принимает количество секунд (например, 1, 30, 60) или значение с суффиксом времени (например, `30s`, `1m`, `1h`). Возможные значения: 0-1h. По умолчанию: 10s.
message_format	integer	Формат сообщения. Возможные значения: 0 - Обычный текст; 1 - *(по умолчанию)* HTML.
script	text	Тело скрипта вебхука (JavaScript). Поведение свойства: - *поддерживается*, если для `type` установлено значение "Email" Поведение свойства: - *обязательно*, если для `type` установлено значение "Webhook"

Свойство	Тип	Описание
timeout	string	Тайм-аут скрипта вебхука. Принимает количество секунд (например, 1, 60) или значение с суффиксом времени (например, 30s, 1m) Возможные значения: 1-60s. По умолчанию: 30s. Поведение свойства: - *поддерживается*, если для type установлено значение "Webhook"
process_tags	integer	Обрабатывать значения свойств JSON в ответе скрипта вебхука как теги. Эти теги добавляются к существующим тегам проблемы. Возможные значения: 0 - *(по умолчанию)* Игнорировать ответ скрипта вебхука; 1 - Обрабатывать ответ скрипта вебхука как теги. Поведение свойства: - *поддерживается*, если для type установлено значение "Webhook"
show_event_menu	integer	Добавлять запись в меню событий, содержащую ссылку на пользовательский URL. Также добавляет свойство urls в результат problem.get и event.get. Возможные значения: 0 - *(по умолчанию)* Не добавлять запись в меню событий или свойство urls; 1 - Добавлять запись в меню событий и свойство urls. Поведение свойства: - *поддерживается*, если для type установлено значение "Webhook"
event_menu_url	string	URL, используемый в записи меню событий и свойстве urls, возвращаемом методами problem.get и event.get. Поведение свойства: - *поддерживается*, если для show_event_menu установлено значение "Include event menu entry and urls property"
event_menu_name	string	Название, используемое для записи меню событий и в свойстве urls, возвращаемом методами problem.get и event.get. Поведение свойства: - *поддерживается*, если для show_event_menu установлено значение "Include event menu entry and urls property"
parameters	array	Параметры вебхука или скрипта. Поведение свойства: - *поддерживается*, если для type установлено значение "Webhook" или "Script"
description	text	Описание типа носителя.

Параметры вебхука

Параметры вебхука имеют следующие свойства.

Property	Type	Description
name	string	Имя параметра. Поведение свойства: - *required*
value	string	Значение параметра, поддерживает макросы. Поддерживаемые макросы описаны на странице Поддерживаемые макросы.

Параметры скрипта

Параметры скрипта имеют следующие свойства.

Property	Type	Description
sortorder	integer	Порядок, в котором значения параметров будут передаваться скрипту как аргументы командной строки, начиная с 0 как первого. Property behavior: - *required*
value	string	Значение параметра, поддерживает макросы. Поддерживаемые макросы описаны на странице Supported macros.

Шаблон сообщения

Объект шаблона сообщения определяет шаблон, который будет использоваться как сообщение по умолчанию для операций действия при отправке уведомления. Он имеет следующие свойства.

Property	Type	Description
eventsource	integer	Источник события. Возможные значения: 0 - Триггеры; 1 - Обнаружение; 2 - Автоматическая регистрация; 3 - Внутренние; 4 - Сервисы. Поведение свойства: - *required*
recovery	integer	Режим операции. Возможные значения: 0 - Операции; 1 - Операции восстановления; 2 - Операции обновления. Поведение свойства: - *required*
subject	string	Тема сообщения.
message	string	Текст сообщения.

mediatype.create

Описание

`object mediatype.create(object/array mediaTypes)`

Этот метод позволяет создавать новые типы носителей.

Note:
Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.
Если способы доставки указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса, пользователь не может вызвать этот метод для соответствующих типов носителей.

Параметры

`(object/array)` Типы оповещений для создания.

В дополнение к стандартным свойствам типа оповещений, метод принимает следующие параметры.

Parameter	Type	Description
message_templates	array	Шаблоны сообщений, которые будут созданы для типа оповещений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных способов оповещения, указанных в свойстве`mediatypeids`. Порядок возвращаемых ID совпадает с порядком переданных способов оповещения.

Примеры

Создание способа оповещения по электронной почте.

Создать новый способ оповещения по электронной почте с настраиваемым SMTP портом и шаблонами сообщений.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.R
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Создание способа оповещения с использованием скрипта

Создать новый способ оповещения с использованием скрипта с пользовательским значением для количества попыток и интервала между ними.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Создание способа оповещения с использованием вебхука

Создать новый способ оповещения с использованием вебхука.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "4",
```

```
        "name": "Webhook",
        "script": "var Webhook = {\r\n    token: null,\r\n    to: null,\r\n    subject: null,\r\n    messa
        "parameters": [
            {
                "name": "Message",
                "value": "{ALERT.MESSAGE}"
            },
            {
                "name": "Subject",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "name": "To",
                "value": "{ALERT.SENDTO}"
            },
            {
                "name": "Token",
                "value": "<Token>"
            }
        ]
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```


Источник

CMediaType::create() в *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Описание

`object mediatype.delete(array mediaTypeIds)`

Этот метод позволяет удалить типы средств оповещения.

> **Note:**
> Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
 Если способы доставки указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса, пользователь не может вызвать этот метод для соответствующих типов средств оповещения.

Параметры

( ) ID удаляемых способов оповещения.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных способов оповещения, указанных в свойстве `mediatypeids`.

Примеры

Удаление нескольких способов оповещения

Удалить два способа оповещения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Источник

CMediaType::delete() в *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Описание

`integer/array mediatype.get(object parameters)`

Метод позволяет получить типы носителей согласно указанным параметрам.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
 Если способы доставки указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса, пользователь не может вызвать этот метод для соответствующих типов носителей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Note:
При запросе информации, связанной с пользователями, о типах медиа *Admin* и *User* могут получать данные только о своем пользователе. Пример см. в разделе Получение типов медиа как *Admin*.

Parameter	Type	Description
mediatypeids	ID/array	Возвращать только типы медиа с указанными ID.
mediaids	ID/array	Возвращать только типы медиа, используемые указанными media.
userids	ID/array	Возвращать только типы медиа, используемые указанными пользователями.
selectActions	query	Возвращать свойство `actions` с действиями, которые используют тип медиа.

Parameter	Type	Description
selectMessageTemplates	query	Возвращать свойство `message_templates` с массивом сообщений типа медиа. Поведение параметра: - *supported* для пользователей типа *Super admin*
selectUsers	query	Возвращать свойство `users` с пользователями, которые используют тип медиа. См. `user.get` для ограничений в зависимости от типа пользователя.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `mediatypeid`.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сравнения. Не поддерживает свойства с типом данных `text`.
output	query	Возможные свойства объекта типа медиа для пользователей типа *Admin* и *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`. Свойства объекта типа медиа, которые нужно вернуть. Пользователи типа *Admin* и *User* могут получать только следующие свойства: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Пример см. в разделе Получение типов медиа как *Admin*. По умолчанию: `extend`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств, а значения — строки для поиска. Если дополнительные параметры не указаны, будет выполнен поиск `LIKE "%…%"`. Поддерживает только свойства с типом данных `string` и `text`. Возможные свойства объекта типа медиа для пользователей типа *Admin* и *User*: `name`, `description`.
countOutput	boolean	Эти параметры описаны в комментарии к справочнику.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если был использован параметр `countOutput`.

Примеры

Получение типов оповещений

Получите все настроенные типы оповещений. Следующий пример возвращает два типа оповещений:

- тип оповещений по электронной почте;
- тип оповещений SMS.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "smtp_server": "mail.example.com",
            "smtp_helo": "example.com",
            "smtp_email": "zabbix@example.com",
            "exec_path": "",
            "gsm_modem": "",
            "username": "",
            "passwd": "",
            "status": "0",
            "smtp_port": "25",
            "smtp_security": "0",
            "smtp_verify_peer": "0",
            "smtp_verify_host": "0",
            "smtp_authentication": "0",
            "maxsessions": "1",
            "maxattempts": "3",
            "attempt_interval": "10s",
            "message_format": "0",
            "script": "",
            "timeout": "30s",
            "process_tags": "0",
            "show_event_menu": "1",
            "event_menu_url": "",
            "event_menu_name": "",
            "description": "",
            "provider": "0",
            "message_templates": [
                {
                    "eventsource": "0",
                    "recovery": "0",
                    "subject": "Problem: {EVENT.NAME}",
                    "message": "Problem started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "Resolved: {EVENT.NAME}",
                    "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE
                },
                {
```

```
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.
            },
            {
                "eventsource": "2",
                "recovery": "0",
                "subject": "Autoregistration: {HOST.HOST}",
                "message": "Host name: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.PORT}"
            }
        ],
        "parameters": []
    },
    {
        "mediatypeid": "3",
        "type": "2",
        "name": "SMS",
        "smtp_server": "",
        "smtp_helo": "",
        "smtp_email": "",
        "exec_path": "",
        "gsm_modem": "/dev/ttyS0",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "1",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "",
                "message": "{EVENT.SEVERITY}: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "",
                "message": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
            },
```

```
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "",
                    "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "",
                    "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "",
                    "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
                }
            ],
            "parameters": []
        }
    ],
    "id": 1
}
```


Получение способов оповещения от имени *Admin*

Пользователь типа *Admin* может получить все включённые способы оповещения вместе с пользователями, которые используют эти способы оповещения. Следующий пример возвращает два способа оповещения:

- способ оповещения email с одним пользователем (только собственный пользователь пользователя типа *Admin*);
- способ оповещения SMS без пользователей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
                    "surname": "Doe",
```

```
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```


Получение способов оповещения с использованием скриптов и вебхуков

В следующем примере возвращаются три способа оповещения:

- способ оповещения с использованием скрипта с параметрами;
- способ оповещения с использованием скрипта без параметров;
- способ оповещения с использованием вебхука с параметрами.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
```

```
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```


Смотрите также

- Пользователь

Источник

CMediaType::get() в *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Описание

`object mediatype.update(object/array mediaTypes)`

Этот метод позволяет обновлять существующие типы носителей.

> **Note:**
> Этот метод доступен только пользователям типа *Суперадминистратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователей. Дополнительную информацию см. в разделе Роли пользователей.
 Если способы доставки указаны в функциональном флаге `$ZBX_FEATURE_FLAGS['media_type_denylist']` в файле конфигурации веб-интерфейса, пользователь не может вызывать этот метод для соответствующих типов носителей.

Параметры

`(object/array)` Свойства типа оповещений, которые необходимо обновить.

Для каждого типа оповещений должно быть определено свойство `mediatypeid`, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Помимо стандартных свойств типа оповещений, метод принимает следующие параметры.

Parameter	Type	Description
message_templates	array	Шаблоны сообщений для замены текущих шаблонов сообщений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных способов оповещений, указанных в свойстве `mediatypeids`.

Примеры

Включение типа медиа

Включите тип медиа, то есть установите для него статус ”0”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Источник

CMediaType::update() в *ui/include/classes/api/services/CMediaType.php*.

Трассировка APM

Этот класс предназначен для работы с трассировками, собранными для мониторинга производительности приложений (APM).

Ссылки на объекты:

- Трассировка APM
- Событие

- Связь

Доступные методы:

- apmtrace.get - получение трассировок

Объект трассировки APM

Следующие объекты напрямую связаны с API `apmtrace`.

Note:
Каждое свойство, за исключением `span_count` и `error_count`, соответствует столбцу в таблице `otel_traces` источника данных телеметрии. Большинство свойств соответствуют столбцу с таким же именем (например, свойство `span_name` соответствует столбцу `SpanName`). Если имена различаются, столбцы источника данных указаны в таблицах ниже. Список столбцов см. в разделе Поддерживаемые столбцы данных телеметрии.

Трассировка APM

Объект трассировки APM представляет трассировку (путь, который запрос проходит через ваши приложения и службы). Трассировка состоит из одного или нескольких интервалов; каждый интервал представляет одну операцию.

Объект трассировки APM основан на корневом интервале трассировки (интервале, с которого начинается трассировка и у которого нет родительского интервала). Все свойства содержат данные корневого интервала, за исключением `span_count` и `error_count`, которые вычисляются на основе всех интервалов трассировки.

Объект трассировки APM имеет следующие свойства:

Свойство	Тип	Описание
timestamp	integer	Время Unix (в наносекундах), когда начался корневой интервал. Поведение свойства: - *только для чтения*
traceid	string	Идентификатор трассировки, общий для всех интервалов, относящихся к трассировке. Поведение свойства: - *только для чтения*
spanid	string	Идентификатор корневого интервала. Поведение свойства: - *только для чтения*
trace_state	string	Дополнительные данные, если они есть, которые инструменты трассировки добавляют к трассировке (см. контекст трассировки W3C). Поведение свойства: - *только для чтения*
span_name	string	Имя операции, которую представляет корневой интервал (например, `POST /zabbix.php`). Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
span_kind	string	Роль корневого интервала. Возможные значения: `Unspecified` - тип интервала не указан; `Internal` - операция, выполняемая внутри приложения; `Server` - обработка входящего запроса; `Client` - исходящий запрос; `Producer` - создание асинхронной операции (например, отправка сообщения); `Consumer` - обработка асинхронной операции. Поведение свойства: - *только для чтения*
service_name	string	Имя службы, создавшей корневой интервал (например, `zabbix-nginx`), полученное из атрибута ресурса OpenTelemetry `service.name`. Поведение свойства: - *только для чтения*
resource_attributes	object	Пары ключ-значение атрибутов ресурса корневого интервала (например, `host.name: eabf8c2a5ddc`). Поведение свойства: - *только для чтения*
scope_name	string	Имя области инструментирования, создавшей корневой интервал. Поведение свойства: - *только для чтения*
scope_version	string	Версия области инструментирования. Поведение свойства: - *только для чтения*
span_attributes	object	Пары ключ-значение атрибутов интервала корневого интервала (например, `http.method: POST`). Поведение свойства: - *только для чтения*
duration	integer	Продолжительность трассировки в наносекундах. Поведение свойства: - *только для чтения*
status_code	string	Статус операции, которую представляет корневой интервал. Возможные значения: `Unset` - корневому интервалу не присвоен явный статус; `Ok` - операция успешно завершена; `Error` - операция завершена с ошибкой. Поведение свойства: - *только для чтения*
status_message	string	Дополнительная информация, если она есть, о статусе корневого интервала. Поведение свойства: - *только для чтения*
events	array	Массив объектов событий корневого интервала, то есть значимых моментов времени во время выполнения операции (например, исключений). Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
links	array	Массив объектов связей корневого интервала, указывающих на связанные интервалы в той же или другой трассировке. Поведение свойства: - *только для чтения*
span_count	integer	Общее количество интервалов в трассировке. Поведение свойства: - *только для чтения*
error_count	integer	Количество интервалов в трассировке, для которых `status_code` имеет значение `Error`. Поведение свойства: - *только для чтения*

Событие

Объект события имеет следующие свойства.

Property	Type	Description
timestamp	integer	Время в формате Unix (в наносекундах), когда произошло событие. Столбец источника данных: `Events.Timestamp`
name	string	Имя события. Столбец источника данных: `Events.Name`
attributes	object	Атрибуты «ключ-значение» с дополнительной информацией о событии, если она есть. Столбец источника данных: `Events.Attributes`

Ссылка

Объект ссылки имеет следующие свойства.

Свойство	Тип	Описание
traceid	string	Идентификатор трассировки, содержащей связанную операцию. Столбец источника данных: `Links.TraceId`
spanid	string	Идентификатор связанной операции. Вместе с `traceid` однозначно идентифицирует связанную операцию. Столбец источника данных: `Links.SpanId`
trace_state	string	Состояние трассировки связанной операции, если оно задано (см. W3C Trace Context). Столбец источника данных: `Links.TraceState`
attributes	object	Атрибуты «ключ-значение» с дополнительной информацией о ссылке. Столбец источника данных: `Links.Attributes`

apmtrace.get

Описание

`integer/array apmtrace.get(object parameters)`

Метод позволяет получать трассировки в соответствии с указанными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
time_from	timestamp	Возвращать только трассировки, корневой спан которых начался в указанное время или позже. Поведение параметра: - *обязательный*
time_till	timestamp	Возвращать только трассировки, корневой спан которых начался до указанного времени. Поведение параметра: - *обязательный*
traceids	string/array	Возвращать только трассировки с указанными идентификаторами.
resource_attributes	object/array	Возвращать только трассировки, корневой спан которых имеет указанные атрибуты ресурса. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все трассировки. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
resource_attributes_evaltype	integer	Метод оценки условий в `resource_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
span_attributes	object/array	Возвращать только трассировки, корневой спан которых имеет указанные атрибуты спана. Формат: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все трассировки. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
span_attributes_evaltype	integer	Метод оценки условий в `span_attributes`. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
min_duration	integer	Возвращать только трассировки, у которых `duration` корневого спана больше или равно указанному значению в наносекундах.
max_duration	integer	Возвращать только трассировки, у которых `duration` корневого спана меньше указанного значения в наносекундах.

Параметр	Тип	Описание
filter	object	Возвращать только результаты, которые точно соответствуют указанному фильтру. Принимает объект, в котором ключами являются имена свойств, а значениями - либо отдельное значение, либо массив значений для сопоставления. Поддерживаемые свойства: `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, в котором ключами являются имена свойств, а значениями - строки для поиска. Если дополнительные параметры не указаны, выполняется поиск `LIKE "%…%"`. Поддерживаемые свойства: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Количество записей, которые необходимо пропустить перед возвратом результата.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение медленных трассировок

Получите две самые медленные трассировки сервиса `zabbix-nginx`. Выполните поиск трассировок с 10:00 до 11:00 1 октября 2026 года (UTC) и включите только те, выполнение которых заняло 1 секунду или больше.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
```

```
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```

См. также

- APM span
- APM metric
- APM log

Источник

CApmTrace::get() в *ui/include/classes/api/services/CApmTrace.php*.

Тренд

Этот класс предназначен для работы с данными трендов.

Ссылки на объекты:

- Тренд с плавающей точкой
- Целочисленный тренд

Доступные методы:

- trend.get - получить тренды

Объект trend

Следующие объекты напрямую связаны с API `trend`.

Note:
Объекты trend различаются в зависимости от типа информации элемента данных. Они создаются сервером Zabbix и не могут быть изменены через API.

Объект тренда с плавающей точкой

Объект тренда с плавающей точкой имеет следующие свойства.

Property	Type	Description
clock	timestamp	Метка времени часа, для которого было вычислено значение. Например, метка времени ”04:00:00” означает значения, вычисленные за период ”04:00:00-04:59:59”.
itemid	ID	ID связанного элемента данных.
num	integer	Количество значений, доступных за час.
value_min	float	Минимальное значение за час.
value_avg	float	Среднее значение за час.
value_max	float	Максимальное значение за час.

Целочисленный тренд

Объект целочисленного тренда имеет следующие свойства.

Property	Type	Description
clock	timestamp	Метка времени часа, для которого было рассчитано значение. Например, метка времени ”04:00:00” означает значения, рассчитанные за период ”04:00:00-04:59:59”.
itemid	ID	ID связанного элемента данных.
num	integer	Количество значений, которые были доступны за час.
value_min	integer	Минимальное значение за час.
value_avg	integer	Среднее значение за час.
value_max	integer	Максимальное значение за час.

trend.get

Описание

`integer/array trend.get(object parameters)`

Метод позволяет получать данные трендов в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешение на вызов метода может быть отозвано в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
itemids	ID/array	Возвращать только тренды с указанными ID элементов данных.
time_from	timestamp	Возвращать только значения, собранные после указанного времени или в указанное время.
time_till	timestamp	Возвращать только значения, собранные до указанного времени или в указанное время.
countOutput	boolean	Подсчитать количество полученных объектов.
limit	integer	Ограничить количество полученных объектов.
output	query	Задать свойства объекта Trend, которые нужно вернуть.

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение данных трендов элемента данных

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

Источник

CTrend::get() в *ui/include/classes/api/services/CTrend.php*.

Триггер

Этот класс предназначен для работы с триггерами.

Ссылки на объекты:

- Trigger
- Trigger tag

Доступные методы:

- trigger.create - создать новые триггеры
- trigger.delete - удалить триггеры
- trigger.get - получить триггеры
- trigger.update - обновить триггеры

Объект триггера

Следующие объекты напрямую связаны с API `trigger`.

Триггер

Объект триггера имеет следующие свойства.

Property	Type	Description
triggerid	ID	ID триггера. Property behavior: - *read-only* - *required* for update operations
description	string	Имя триггера. Property behavior: - *required* for create operations
expression	string	Выражение триггера. Элементы данных с `value_type`, установленным в 5 (binary) или 6 (JSON), не могут использоваться в выражениях триггера. Property behavior: - *required* for create operations
event_name	string	Имя события, создаваемого триггером.
opdata	string	Операционные данные.
comments	string	Дополнительное описание триггера.
error	string	Текст ошибки, если при обновлении состояния триггера возникли какие-либо проблемы. Property behavior: - *read-only*
flags	integer	Происхождение триггера. Возможные значения: 0 - *(default)* обычный триггер; 4 - триггер, преобразованный из прототипа. Property behavior: - *read-only*
lastchange	timestamp	Время последнего изменения состояния триггера. Property behavior: - *read-only*
priority	integer	Серьезность триггера. Возможные значения: 0 - *(default)* не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.
state	integer	Состояние выражения триггера. Возможные значения: 0 - *(default)* состояние триггера актуально; 1 - текущее состояние триггера неизвестно. Property behavior: - *read-only*
status	integer	Включен или отключен триггер. Возможные значения: 0 - *(default)* включен; 1 - отключен.

Property	Type	Description
templateid	ID	ID родительского триггера шаблона. Property behavior: - *read-only*
type	integer	Может ли триггер генерировать несколько событий проблемы. Возможные значения: 0 - *(default)* не генерировать несколько событий; 1 - генерировать несколько событий.
url	string	URL, связанный с триггером.
url_name	string	Метка для URL, связанного с триггером.
value	integer	Находится ли триггер в состоянии OK или проблемы. Возможные значения: 0 - *(default)* OK; 1 - проблема. Property behavior: - *read-only*
recovery_mode	integer	Режим генерации события OK. Возможные значения: 0 - *(default)* Expression; 1 - Recovery expression; 2 - None.
recovery_expression	string	Выражение восстановления триггера. Элементы данных с `value_type`, установленным в 5 (binary) или 6 (JSON), не могут использоваться в выражениях восстановления триггера.
correlation_mode	integer	Закрытие события OK. Возможные значения: 0 - *(default)* All problems; 1 - All problems if tag values match.
correlation_tag	string	Тег для сопоставления.
manual_close	integer	Разрешить ручное закрытие. Возможные значения: 0 - *(default)* Нет; 1 - Да.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных триггеров с уже существующими. Если не указан, генерируется автоматически. Property behavior: - *supported* if the trigger belongs to a template

Тег триггера

Объект тега триггера имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега триггера. Поведение свойства: - *обязательное*

Property	Type	Description
value	string	Значение тега триггера. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети; 3 - элемент данных. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.
automatic	integer	Тип тега триггера: Возможные значения: 0 - *(по умолчанию)* тег, созданный пользователем; 1 - тег, созданный LLD;Поведение свойства: - *только для чтения*.

trigger.create

Описание

`object trigger.create(object/array triggers)`

Этот метод позволяет создавать новые триггеры.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Триггеры для создания.

В дополнение к стандартным свойствам триггера метод принимает следующие параметры.

Parameter	Type	Description
dependencies	array	Триггеры , от которых зависит триггер. У триггеров должно быть определено только свойство `triggerid`.
tags	array	Теги триггера.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных триггеров, указанных в свойстве`triggerids`. Порядок возвращаемых ID совпадает с порядком переданных триггеров.

Примеры

Создание триггера

Создайте два триггера, каждый из которых зависит от другого триггера.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "                          {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "          ",
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
                {
                    "triggerid": "17368"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Источник

CTrigger::create() в *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Описание

`object trigger.delete(array triggerIds)`

Этот метод позволяет удалять триггеры.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе User roles.

Параметры

( ) ID удаляемых триггеров.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных триггеров, указанных в свойстве `triggerids`.

Примеры

Удаление нескольких триггеров

Удаление двух триггеров.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Источник

CTrigger::delete() в *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Описание

`integer/array trigger.get(object parameters)`

Метод позволяет получать триггеры в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
triggerids	ID/array	Возвращать только триггеры с указанными идентификаторами.
groupids	ID/array	Возвращать только триггеры, принадлежащие узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
templateids	ID/array	Возвращать только триггеры, принадлежащие указанным шаблонам.

Параметр	Тип	Описание
hostids	ID/array	Возвращать только триггеры, принадлежащие указанным узлам сети.
itemids	ID/array	Возвращать только триггеры, содержащие указанные элементы данных.
maintenanceids	ID/array	Возвращать только триггеры, на которые влияют указанные периоды обслуживания.
functions	string/array	Возвращать только триггеры, использующие указанные функции. Список поддерживаемых функций приведён на странице поддерживаемых функций.
group	string	Возвращать только триггеры, принадлежащие узлам сети или шаблонам из группы узлов сети или группы шаблонов с указанным именем.
host	string	Возвращать только триггеры, принадлежащие узлу сети с указанным техническим именем.
inherited	boolean	Если установлено значение `true`, возвращать только триггеры, унаследованные от шаблона.
templated	boolean	Если установлено значение `true`, возвращать только триггеры, принадлежащие шаблонам.
dependent	boolean	Если установлено значение `true`, возвращать только триггеры, имеющие зависимости. Если установлено значение `false`, возвращать только триггеры, не имеющие зависимостей.
monitored	flag	Возвращать только включённые триггеры, принадлежащие отслеживаемым узлам сети и содержащие только включённые элементы данных.
active	flag	Возвращать только включённые триггеры, принадлежащие отслеживаемым узлам сети.
maintenance	boolean	Если установлено значение `true`, возвращать только включённые триггеры, принадлежащие узлам сети, находящимся на обслуживании.
withUnacknowledgedEvents	flag	Возвращать только триггеры, имеющие неподтверждённые события.
withAcknowledgedEvents	flag	Возвращать только триггеры, у которых все события подтверждены.
withLastEventUnacknowledged	flag	Возвращать только триггеры с неподтверждённым последним событием.
skipDependent	flag	Пропускать триггеры в состоянии проблемы, зависящие от других триггеров. Обратите внимание, что другие триггеры игнорируются, если они отключены, имеют отключённые элементы данных или принадлежат узлам сети с отключёнными элементами данных.
lastChangeSince	timestamp	Возвращать только триггеры, изменившие своё состояние после указанного времени.
lastChangeTill	timestamp	Возвращать только триггеры, изменившие своё состояние до указанного времени.
only_true	flag	Возвращать только триггеры, которые недавно находились в состоянии проблемы (в течение `ok_period`).
min_severity	integer	Возвращать только триггеры с важностью, большей или равной указанной.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.

Параметр	Тип	Описание
tags	array	Возвращать только триггеры с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все триггеры. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
expandComment	flag	Раскрывать макросы в описании триггера.
expandDescription	flag	Раскрывать макросы в имени триггера.
expandExpression	flag	Раскрывать функции и макросы в выражении триггера.
selectHostGroups	query	Возвращать группы узлов сети, которым принадлежит триггер, в свойстве `hostgroups`.
selectHosts	query	Возвращать узлы сети, которым принадлежит триггер, в свойстве `hosts`.
selectItems	query	Возвращать элементы данных, содержащиеся в триггере, в свойстве `items`.
selectFunctions	query	Возвращать функции, используемые в триггере, в свойстве `functions`. Объекты функций представляют функции, используемые в выражении триггера, и имеют следующие свойства: `functionid` - (ID) идентификатор функции; `itemid` - (ID) идентификатор элемента данных, используемого в функции; `function` - (string) имя функции; `parameter` - (string) параметр, переданный функции. Параметр запроса заменяется символом $ в возвращаемой строке.
selectDependencies	query	Возвращать триггеры, от которых зависит данный триггер, в свойстве `dependencies`.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения триггера. Объект обнаружения триггера связывает обнаруженный триггер с прототипом триггера, на основе которого он был обнаружен. Он имеет следующие свойства: `parent_triggerid` - (ID) идентификатор прототипа триггера, на основе которого был создан триггер; `status` - (int) статус обнаружения триггера: 0 - *(по умолчанию)* триггер обнаружен, 1 - триггер больше не обнаруживается; `ts_delete` - (timestamp) время удаления триггера, который больше не обнаруживается; `ts_disable` - (timestamp) время отключения триггера, который больше не обнаруживается; `disable_source` - (int) признак того, был ли триггер отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.

Параметр	Тип	Описание
selectDiscoveryRule	query	Возвращать правило обнаружения низкого уровня, создавшее триггер, в свойстве `discoveryRule`.
selectLastEvent	query	Возвращать последнее значимое событие триггера в свойстве `lastEvent`.
selectTags	query	Возвращать теги триггера в свойстве `tags`.
inheritedTags	boolean	Возвращать триггеры, имеющие указанные `tags` также в шаблоне, узле сети или связанных шаблонах. Возможные значения: `true` - шаблон, узел сети или связанные шаблоны также должны иметь указанные теги; `false` - *(по умолчанию)* теги из шаблона, узла сети или связанных шаблонов игнорируются.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, унаследованными от шаблонов, связанных шаблонов и узлов сети, а также от элементов данных, на которые ссылаются выражения проблемы или восстановления триггера.
selectTemplateGroups	query	Возвращать группы шаблонов, которым принадлежит триггер, в свойстве `templategroups`.
filter	object	Возвращать только результаты, точно соответствующие указанному фильтру. Принимает объект, в котором ключами являются имена свойств, а значениями - отдельное значение или массив значений для сопоставления. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, которому принадлежит триггер; `hostid` - идентификатор узла сети, которому принадлежит триггер.
limitSelects	integer	Ограничивает количество записей, возвращаемых вложенными выборками. Применяется к следующим вложенным выборкам: `selectHosts` - результаты сортируются по `host`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Эти параметры описаны в справочных комментариях.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	

Параметр	Тип	Описание
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Возвращать объект обнаружения триггера в свойстве `triggerDiscovery`. Объекты обнаружения триггеров связывают триггер с прототипом триггера, на основе которого он был создан. Этот запрос **устарел**; вместо него используйте `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение данных по ID триггера

Получите все данные и функции, используемые в триггере ”14062”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "14062",
            "expression": "{13513}<10m",
            "description": "{HOST.NAME}           (        < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "                   10    ",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
```

```
            "opdata": "",
            "event_name": "",
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "triggerid": "14062",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Получение триггеров в состоянии проблемы

Получите ID, имя и уровень серьезности всех триггеров в состоянии проблемы и отсортируйте их по уровню серьезности по убыванию.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Получение конкретного триггера с тегами

Получение конкретного триггера с тегами.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}",
                    "automatic": "1"
                },
                {
                    "tag": "error",
                    "value": "",
                    "automatic": "1"
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Правило обнаружения
- Элемент данных
- Узел сети
- Группа узлов сети
- Группа шаблонов

Источник

CTrigger::get() в *ui/include/classes/api/services/CTrigger.php*.

trigger.update

Описание

`object trigger.update(object/array triggers)`

Этот метод позволяет обновлять существующие триггеры.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Свойства триггера, которые необходимо обновить.

Для каждого триггера должно быть определено свойство triggerid, все остальные свойства необязательны. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Дополнительно к стандартным свойствам триггера метод принимает следующие параметры.

Parameter	Type	Description
dependencies	array	Триггеры, от которых зависит триггер. Для триггеров должно быть определено только свойство triggerid.
tags	array	Теги триггера.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы обновленных триггеров в свойстве triggerids.

Примеры

Включение триггера

Включите триггер, то есть установите для него статус ”0”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Замена тегов триггера

Замените теги для триггера.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "tags": [
            {
                "tag": "service",
                "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
            },
            {
                "tag": "error",
                "value": ""
```

```
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Замена зависимостей

Замените зависимости для триггера.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Источник

CTrigger::update() в *ui/include/classes/api/services/CTrigger.php*.

Узел высокой доступности

Этот класс предназначен для работы с узлами сервера, которые входят в кластер высокой доступности, или с отдельным экземпляром сервера.

Ссылки на объекты:

- Узел высокой доступности

Доступные методы:

- hanode.get - получить узлы

Объект узла высокой доступности

Следующий объект связан с эксплуатацией кластера высокой доступности серверов Zabbix.

Узел высокой доступности

Note:
Узлы создаются сервером Zabbix и не могут быть изменены через API.

Объект узла высокой доступности имеет следующие свойства.

Property	Type	Description
ha_nodeid	ID	ID узла.
name	string	Имя, назначенное узлу, с использованием параметра конфигурации HANodeName в `zabbix_server.conf`. Пусто для сервера, работающего в автономном режиме.
address	string	IP-адрес или DNS-имя, с которого узел подключается.
port	integer	Порт, на котором работает узел.
lastaccess	integer	Время heartbeat, то есть время последнего обновления от узла. Метка времени UTC.
status	integer	Состояние узла. Возможные значения: 0 - standby; 1 - остановлен вручную; 2 - недоступен; 3 - active.

hanode.get

Описание

`integer/array hanode.get(object parameters)`

Метод позволяет получить список узлов кластера высокой доступности в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типов пользователей *Super admin*. Дополнительную информацию см. в разделе User roles.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
ha_nodeids	ID/array	Возвращать только узлы сети с указанными ID узлов сети.
filter	object	Возвращать только те результаты, которые в точности соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сопоставления. Поддерживаемые свойства: `name`, `address`, `status`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `name`, `lastaccess`, `status`.
countOutput	flag	Эти параметры описаны в справочном комментарии.
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
sortorder	string/array	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получить список узлов сети, отсортированных по статусу

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",
      "address": "192.168.1.6",
      "port": "10051",
      "lastaccess": "1635335705",
      "status": "0"
```

```
    }
  },
  "id": 1
}
```

Получить список конкретных узлов по их ID

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "name": "node4",
            "address": "192.168.1.8",
            "port": "10052",
            "lastaccess": "1635334214",
            "status": "1"
        },
        {
            "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
            "name": "node6",
            "address": "192.168.1.10",
            "port": "10053",
            "lastaccess": "1635332902",
            "status": "2"
        }
    ],
    "id": 1
}
```

Получить список остановленных узлов сети

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
            "address": "192.168.1.12",
```

```
            "port": "10051"
        },
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "address": "192.168.1.8",
            "port": "10052"
        },
        {
            "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
            "address": "192.168.1.9",
            "port": "10053"
        }
    ],
    "id": 1
}
```

Получить количество резервных узлов сети

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Проверка статуса узлов сети по указанным IP-адресам

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
```

```
            "status": "3"
        }
    ],
    "id": 1
}
```

Источник

CHaNode::get() в *ui/include/classes/api/services/CHaNode.php*.

Узел сети

Этот класс предназначен для работы с узлами сети.

Ссылки на объекты:

- Host
- Host inventory
- Host tag

Доступные методы:

- host.create - создать новые узлы сети
- host.delete - удалить узлы сети
- host.get - получить узлы сети
- host.massadd - добавить связанные объекты к узлам сети
- host.massremove - удалить связанные объекты из узлов сети
- host.update - обновить узлы сети

Объект узла сети

Следующие объекты напрямую связаны с API host.

Узел сети

Объект узла сети имеет следующие свойства.

Property	Type	Description
hostid	ID	Идентификатор узла сети. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
host	string	Техническое имя узла сети. Поведение свойства: - *обязательно* для операций создания
description	text	Описание узла сети.
flags	integer	Источник узла сети. Возможные значения: 0 - обычный узел сети; 4 - узел сети, преобразованный из прототипа. Поведение свойства: - *только для чтения*
inventory_mode	integer	Режим заполнения инвентаря узла сети. Возможные значения: -1 - *(по умолчанию)* отключено; 0 - вручную; 1 - автоматически.

Property	Type	Description
ipmi_authtype	integer	Алгоритм аутентификации IPMI. Возможные значения: -1 - *(по умолчанию)* по умолчанию; 0 - отсутствует; 1 - MD2; 2 - MD5 4 - прямой; 5 - OEM; 6 - RMCP+.
ipmi_password	string	Пароль IPMI.
ipmi_privilege	integer	Уровень привилегий IPMI. Возможные значения: 1 - обратный вызов; 2 - *(по умолчанию)* пользователь; 3 - оператор; 4 - администратор; 5 - OEM.
ipmi_username	string	Имя пользователя IPMI.
maintenance_from	timestamp	Время начала действующего обслуживания. Поведение свойства: - *только для чтения*
maintenance_status	integer	Статус действующего обслуживания. Возможные значения: 0 - *(по умолчанию)* обслуживание отсутствует; 1 - обслуживание выполняется. Поведение свойства: - *только для чтения*
maintenance_type	integer	Тип действующего обслуживания. Возможные значения: 0 - *(по умолчанию)* обслуживание со сбором данных; 1 - обслуживание без сбора данных. Поведение свойства: - *только для чтения*
maintenanceid	ID	Идентификатор обслуживания, которое в данный момент выполняется на узле сети. Поведение свойства: - *только для чтения*
name	string	Отображаемое имя узла сети. По умолчанию: значение свойства `host`.
monitored_by	integer	Источник, используемый для мониторинга узла сети. Возможные значения: 0 - *(по умолчанию)* сервер Zabbix; 1 - прокси; 2 - группа прокси. Поведение свойства: - *обязательно*, если `select_server_for_monitoring` установлено в значение «Отключено».

Property	Type	Description
proxyid	ID	Идентификатор прокси, используемого для мониторинга узла сети. Поведение свойства: - *обязательно*, если `monitored_by` установлено в значение «Прокси»
proxy_groupid	ID	Идентификатор группы прокси, используемой для мониторинга узла сети. Поведение свойства: - *обязательно*, если `monitored_by` установлено в значение «Группа прокси»
status	integer	Статус и функция узла сети. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
tls_connect	integer	Подключения к узлу сети. Возможные значения: 1 - *(по умолчанию)* без шифрования; 2 - PSK; 4 - сертификат.
tls_accept	integer	Подключения от узла сети. Возможные значения битовой маски: 1 - *(по умолчанию)* без шифрования; 2 - PSK; 4 - сертификат. Это поле является битовой маской; допускается любая сумма возможных значений битовой маски (например, 6 для PSK и сертификата).
tls_issuer	string	Издатель сертификата.
tls_subject	string	Субъект сертификата.
tls_psk_identity	string	Идентификатор PSK; должен быть связан только с одним PSK (в рамках автоматической регистрации, узлов сети и прокси). Не включайте конфиденциальную информацию в идентификатор PSK, поскольку он передается по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK следует использовать. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в значение «PSK» или `tls_accept` содержит бит «PSK»
tls_psk	string	Предварительно установленный ключ (PSK); должен содержать не менее 32 шестнадцатеричных цифр. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в значение «PSK» или `tls_accept` содержит бит «PSK»
active_available	integer	Статус доступности активного интерфейса узла сети. Возможные значения: 0 - статус интерфейса неизвестен; 1 - интерфейс доступен; 2 - интерфейс недоступен. Поведение свойства: - *только для чтения*

Property	Type	Description
assigned_proxyid	ID	Идентификатор прокси, назначенного сервером Zabbix, если мониторинг узла сети выполняется группой прокси. Поведение свойства: *- только для чтения*

Инвентарь узла сети

Объект инвентаря узла сети имеет следующие свойства.

Note:
У каждого свойства есть свой уникальный идентификационный номер, который используется для связывания полей инвентаря узла сети с элементами данных.

ID	Свойство	Тип	Описание	Максимальная длина
4	alias	string	Псевдоним.	128 characters
11	asset_tag	string	Инвентарный номер.	64 characters
28	chassis	string	Шасси.	64 characters
23	contact	string	Контактное лицо.	65535 characters for SQL databases
32	contract_number	string	Номер контракта.	64 characters
47	date_hw_decomm	string	Дата вывода HW из эксплуатации.	64 characters
46	date_hw_expiry	string	Дата окончания обслуживания HW.	64 characters
45	date_hw_install	string	Дата установки HW.	64 characters
44	date_hw_purchase	string	Дата покупки HW.	64 characters
34	deployment_status	string	Статус развертывания.	64 characters
14	hardware	string	Оборудование.	255 characters
15	hardware_full	string	Подробные сведения об оборудовании.	65535 characters for SQL databases
39	host_netmask	string	Маска подсети узла сети.	39 characters
38	host_networks	string	Сети узла сети.	65535 characters for SQL databases
40	host_router	string	Маршрутизатор узла сети.	39 characters
30	hw_arch	string	Архитектура HW.	32 characters
33	installer_name	string	Имя установщика.	64 characters
24	location	string	Местоположение.	65535 characters for SQL databases
25	location_lat	string	Широта местоположения.	16 characters
26	location_lon	string	Долгота местоположения.	16 characters
12	macaddress_a	string	MAC-адрес A.	64 characters
13	macaddress_b	string	MAC-адрес B.	64 characters
29	model	string	Модель.	64 characters
3	name	string	Имя.	128 characters
27	notes	string	Заметки.	65535 characters for SQL databases
41	oob_ip	string	OOB IP-адрес.	39 characters
42	oob_netmask	string	Маска подсети OOB узла сети.	39 characters
43	oob_router	string	OOB маршрутизатор.	39 characters
5	os	string	Имя ОС.	128 characters
6	os_full	string	Подробное имя ОС.	255 characters
7	os_short	string	Краткое имя ОС.	128 characters
61	poc_1_cell	string	Мобильный номер основного POC.	64 characters
58	poc_1_email	string	Основной email.	128 characters
57	poc_1_name	string	Имя основного POC.	128 characters
63	poc_1_notes	string	Заметки основного POC.	65535 characters for SQL databases
59	poc_1_phone_a	string	Телефон A основного POC.	64 characters
60	poc_1_phone_b	string	Телефон B основного POC.	64 characters
62	poc_1_screen	string	Имя в мессенджере основного POC.	64 characters
68	poc_2_cell	string	Мобильный номер вторичного POC.	64 characters
65	poc_2_email	string	Email вторичного POC.	128 characters
64	poc_2_name	string	Имя вторичного POC.	128 characters
70	poc_2_notes	string	Заметки вторичного POC.	65535 characters for SQL databases
66	poc_2_phone_a	string	Телефон A вторичного POC.	64 characters
67	poc_2_phone_b	string	Телефон B вторичного POC.	64 characters

ID	Свойство	Тип	Описание	Максимальная длина
69	poc_2_screen	string	Имя в мессенджере вторичного POC.	64 characters
8	serialno_a	string	Серийный номер A.	64 characters
9	serialno_b	string	Серийный номер B.	64 characters
48	site_address_a	string	Адрес площадки A.	128 characters
49	site_address_b	string	Адрес площадки B.	128 characters
50	site_address_c	string	Адрес площадки C.	128 characters
51	site_city	string	Город площадки.	128 characters
53	site_country	string	Страна площадки.	64 characters
56	site_notes	string	Заметки о площадке.	65535 characters for SQL databases
55	site_rack	string	Расположение стойки на площадке.	128 characters
52	site_state	string	Регион/штат площадки.	64 characters
54	site_zip	string	Почтовый индекс площадки.	64 characters
16	software	string	Программное обеспечение.	255 characters
18	software_app_a	string	Приложение A.	64 characters
19	software_app_b	string	Приложение B.	64 characters
20	software_app_c	string	Приложение C.	64 characters
21	software_app_d	string	Приложение D.	64 characters
22	software_app_e	string	Приложение E.	64 characters
17	software_full	string	Сведения о программном обеспечении.	65535 characters for SQL databases
10	tag	string	Тег.	64 characters
1	type	string	Тип.	64 characters
2	type_full	string	Сведения о типе.	64 characters
35	url_a	string	URL A.	2048 characters
36	url_b	string	URL B.	2048 characters
37	url_c	string	URL C.	2048 characters
31	vendor	string	Поставщик.	64 characters

Тег узла сети

Объект тега узла сети имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега узла сети. Property behavior: - *required*
value	string	Значение тега узла сети. Property behavior: - *read-only*.
automatic	integer	Тип тега узла сети. Possible values: 0 - *(default)* manual (tag created by user); 1 - automatic (tag created by low-level discovery) Property behavior: - *read-only*
object	integer	Тип объекта, от которого был унаследован тег. Possible values: 0 - template. Property behavior: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Property behavior: - *read-only*.

host.create

Описание

`object host.create(object/array hosts)`

Этот метод позволяет создавать новые узлы сети.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Узлы сети для создания.

В дополнение к стандартным свойствам узла сети метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы узлов сети для добавления узла сети. Группы узлов сети должны иметь определенным только свойство `groupid`. Поведение параметра: - *required*
interfaces	object/array	Интерфейсы, которые будут созданы для узла сети.
tags	object/array	Теги узла сети.
templates	object/array	Шаблоны, которые будут привязаны к узлу сети. Шаблоны должны иметь определенным только свойство `templateid`.
macros	object/array	Пользовательские макросы, которые будут созданы для узла сети.
inventory	object	Свойства инвентаря узла сети.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных хостов в свойстве`hostids`. Порядок возвращаемых идентификаторов соответствует порядку переданных хостов.

Примеры

Создание узла сети

Создайте узел сети с именем "Linux server" с IP-интерфейсом и тегами, добавьте его в группу, привяжите к нему шаблон и задайте MAC-адреса в инвентаре узла сети.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
```

```
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Создание узла сети с интерфейсом SNMP

Создайте узел сети под названием "SNMP host", указав интерфейс SNMPv3 и детальную информацию.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "SNMP host",
        "interfaces": [
```

```
            {
                "type": 2,
                "main": 1,
                "useip": 1,
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": 3,
                    "bulk": 0,
                    "securityname": "mysecurityname",
                    "contextname": "",
                    "securitylevel": 1
                }
            }
        ],
        "groups": [
            {
                "groupid": "4"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Создание узла сети с шифрованием PSK

Создайте узел сети с именем "PSK host" и настроенным шифрованием PSK. Обратите внимание, что узел сети должен быть предварительно настроен для использования PSK.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
```

```
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Создание узла сети, контролируемого прокси

Создайте узел сети, который будет контролироваться прокси с ID ”1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Создание узла сети, контролируемого группой прокси

Создайте узел сети, который будет контролироваться группой прокси с ID ”1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "      ,                    ",
        "groups": [
            {
                "groupid": "2"
            }
```

```
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

См. также

- Группа узлов сети
- Шаблон
- Пользовательский макрос
- Интерфейс узла сети
- Инвентарные данные узла сети
- Тег узла сети
- Прокси
- Группа прокси

Источник

CHost::create() в *ui/include/classes/api/services/CHost.php*.

host.delete

Описание

`object host.delete(array hosts)`

Этот метод позволяет удалять узлы сети.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` ID хостов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных хостов под свойством hostids.

Примеры

Удаление нескольких узлов сети

Удалите два узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Источник

CHost::delete() в *ui/include/classes/api/services/CHost.php*.

host.get

Описание

`integer/array host.get(object parameters)`

Метод позволяет получать узлы сети в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
groupids	ID/array	Возвращать только узлы сети, которые принадлежат указанным группам.
dserviceids	ID/array	Возвращать только узлы сети, связанные с указанными обнаруженными службами.
graphids	ID/array	Возвращать только узлы сети, у которых есть указанные графики.
hostids	ID/array	Возвращать только узлы сети с указанными ID узлов сети.
httptestids	ID/array	Возвращать только узлы сети, у которых есть указанные веб-проверки.
interfaceids	ID/array	Возвращать только узлы сети, которые используют указанные интерфейсы.
itemids	ID/array	Возвращать только узлы сети, у которых есть указанные элементы данных.
maintenanceids	ID/array	Возвращать только узлы сети, на которые влияют указанные обслуживания.
monitored_hosts	flag	Возвращать только контролируемые узлы сети.
proxyids	ID/array	Возвращать только узлы сети, которые контролируются указанными прокси.
proxy_groupids	ID/array	Возвращать только узлы сети, которые контролируются указанными группами прокси.
templated_hosts	flag	Возвращать и узлы сети, и шаблоны.
templateids	ID/array	Возвращать только узлы сети, связанные с указанными шаблонами.
triggerids	ID/array	Возвращать только узлы сети, у которых есть указанные триггеры.

Parameter	Type	Description
with_items	flag	Возвращать только узлы сети, у которых есть элементы данных. Переопределяет параметры `with_monitored_items` и `with_simple_graph_items`.
with_item_prototypes	flag	Возвращать только узлы сети, у которых есть прототипы элементов данных. Переопределяет параметр `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Возвращать только узлы сети, у которых есть прототипы элементов данных, разрешенные для создания и имеющие числовой тип информации.
with_graphs	flag	Возвращать только узлы сети, у которых есть графики.
with_graph_prototypes	flag	Возвращать только узлы сети, у которых есть прототипы графиков.
with_httptests	flag	Возвращать только узлы сети, у которых есть веб-проверки. Переопределяет параметр `with_monitored_httptests`.
with_monitored_httptests	flag	Возвращать только узлы сети, у которых есть включенные веб-проверки.
with_monitored_items	flag	Возвращать только узлы сети, у которых есть включенные элементы данных. Переопределяет параметр `with_simple_graph_items`.
with_monitored_triggers	flag	Возвращать только узлы сети, у которых есть включенные триггеры. Все элементы данных, используемые в триггере, также должны быть включены.
with_simple_graph_items	flag	Возвращать только узлы сети, у которых есть элементы данных с числовым типом информации.
with_triggers	flag	Возвращать только узлы сети, у которых есть триггеры. Переопределяет параметр `with_monitored_triggers`.
withProblemsSuppressed	boolean	Если установлено в `true`, возвращать только узлы сети с подавленными проблемами.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И; 2 - ИЛИ.
severities	integer/array	Возвращать узлы сети, у которых есть только проблемы с указанными уровнями серьезности. Применяется только если объект проблемы является триггером.
tags	object/array	Возвращать только узлы сети с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все узлы сети. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
inheritedTags	boolean	Возвращать узлы сети, у которых указанные `tags` также присутствуют во всех связанных с ними шаблонах. Возможные значения: `true` - связанные шаблоны также должны иметь указанные `tags`; `false` - *(по умолчанию)* теги связанных шаблонов игнорируются.

Parameter	Type	Description
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения узла сети. Объект обнаружения узла сети связывает обнаруженный узел сети с прототипом узла сети, из которого он был обнаружен. Он имеет следующие свойства: `host` - `(string)` ID узла сети; `parent_hostid` - `(string)` ID прототипа узла сети, из которого был создан узел сети; `status` - `(int)` статус обнаружения узла сети: 0 - *(по умолчанию)* узел сети обнаружен, 1 - узел сети больше не обнаружен; `ts_delete` - `(timestamp)` время, когда узел сети, который больше не обнаружен, будет удален; `ts_disable` - `(timestamp)` время, когда узел сети, который больше не обнаружен, будет отключен; `disable_source` - `(int)` признак того, был ли узел сети отключен правилом LLD или вручную: 0 - *(по умолчанию)* отключен автоматически, 1 - отключен правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом низкоуровневого обнаружения, которое создало узел сети (из прототипа узла сети в мониторинге VMware).
selectDiscoveryRules	query	Возвращать свойство `discoveryRules` с правилами LLD узла сети. Поддерживает `count`.
selectGraphs	query	Возвращать свойство `graphs` с графиками узла сети. Поддерживает `count`.
selectHostGroups	query	Возвращать свойство `hostgroups` с данными групп узлов сети, к которым принадлежит узел сети.
selectHttpTests	query	Возвращать свойство `httpTests` со сценариями веб-проверок узла сети. Поддерживает `count`.
selectInterfaces	query	Возвращать свойство `interfaces` с интерфейсами узла сети. Поддерживает `count`.
selectInventory	query	Возвращать свойство `inventory` с данными инвентаризации узла сети.
selectItems	query	Возвращать свойство `items` с элементами данных узла сети. Поддерживает `count`.
selectMacros	query	Возвращать свойство `macros` с макросами узла сети.
selectParentTemplates	query	Возвращать свойство `parentTemplates` с шаблонами, к которым привязан узел сети. Помимо полей объекта Template, содержит `link_type` - `(integer)` способ привязки шаблона к узлу сети. Возможные значения: 0 - *(по умолчанию)* привязан вручную; 1 - автоматически привязан LLD. Поддерживает `count`.
selectDashboards	query	Возвращать свойство `dashboards`. Поддерживает `count`.
selectTags	query	Возвращать свойство `tags` с тегами узла сети.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые присутствуют во всех шаблонах, связанных с узлом сети.

Parameter	Type	Description
selectTriggers	query	Возвращать свойство `triggers` с триггерами узла сети. Поддерживает `count`.
selectValueMaps	query	Возвращать свойство `valuemaps` с картами значений узла сети.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи - это имена свойств, а значения - либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства с типом данных `text`. Поддерживает дополнительные свойства: свойства интерфейса узла сети.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectParentTemplates` - результаты будут отсортированы по `host`; `selectInterfaces`; `selectItems` - сортировка по `name`; `selectDiscoveryRules` - сортировка по `name`; `selectTriggers` - сортировка по `description`; `selectGraphs` - сортировка по `name`; `selectDashboards` - сортировка по `name`.
search	object	Возвращать результаты, соответствующие указанному шаблону (без учета регистра). Принимает объект, где ключи - это имена свойств, а значения - строки для поиска. Если дополнительные параметры не указаны, будет выполнен поиск `LIKE "%…%"`. Поддерживает только свойства с типом данных `string` и `text`. Поддерживает дополнительные свойства: свойства интерфейса узла сети.
searchInventory	object	Возвращать узлы сети, у которых данные инвентаризации соответствуют указанному шаблону (без учета регистра). Принимает объект, где ключи - это имена свойств, а значения - строки для поиска. Если дополнительные параметры не указаны, будет выполнен поиск `LIKE "%…%"`. Поддерживает только свойства с типом данных `string` и `text`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Parameter	Type	Description
selectDiscoveries	query	Возвращать свойство `discoveries` с правилами низкоуровневого обнаружения узла сети. Поддерживает `count`. Этот запрос **устарел**, вместо него используйте `selectDiscoveryRules`.
selectHostDiscovery	query	Возвращать свойство `hostDiscovery` с данными объекта обнаружения узла сети. Этот запрос **устарел**, вместо него используйте `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если параметр `countOutput` был использован.

Примеры

Получение данных по имени

Получить все данные о двух узлах сети с именами ”Zabbix server” и ”Linux server”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "proxyid": "0",
            "host": "Zabbix server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Zabbix server",
            "flags": "0",
            "description": "           Zabbix.",
            "tls_connect": "1",
            "tls_accept": "1",
```

```
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```


Получение групп узла сети

Получите группы узла сети, членом которых является узел сети "Zabbix server".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "hostid": "10085",
            "hostgroups": [
                {
                    "groupid": "2",
                    "name": "Linux servers",
                    "flags": "0",
                    "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
                },
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ]
        }
    ],
    "id": 1
}
```

Получение связанных шаблонов

Получите идентификаторы и имена шаблонов, связанных с узлом сети ”10084”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

Получение узлов сети по шаблону

Получите узлы сети, к которым привязан шаблон ”10001” (*Linux by Zabbix agent*).

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Поиск по данным инвентаря узла сети

Получение узлов сети, которые содержат ”Linux” в поле ”OS” инвентаря узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
```

```
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Поиск по тегам узла сети

Получить узлы сети, у которых тег "host-name" равен "linux-server".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Получить узлы сети, у которых эти теги заданы не только на уровне узла сети, но и в связанных родительских шаблонах.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Получить узлы сети, исключая те, у которых есть тег "component".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "tags": [
```

```
                {
                    "tag": "class",
                    "value": "os",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "linux",
                    "automatic": "0"
                }
            ]
        },
        {
            "hostid": "10629",
            "tags": [
                {
                    "tag": "class",
                    "value": "application",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "browser",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Поиск узла сети с тегами и тегами шаблона

Получение узла сети с тегами и всеми тегами, связанными с родительскими шаблонами.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10502",
            "name": "Desktop",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
```

```
                }
            ],
            "inheritedTags": [
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ]
        }
    ],
    "id": 1
}
```

Поиск хостов по серьезности проблемы

Получить хосты, у которых есть проблемы «катастрофа» (Disaster).

Запрос:

```
{
   "jsonrpc": "2.0",
   "method": "host.get",
   "params": {
   "output": ["name"],
   "severities": 5
   },
   "auth": "038e1d7b1735c6a5436ee9eae095879e",
   "id": 1
}
```

Ответ:

```
{
   "jsonrpc": "2.0",
   "result": [
   {
   "hostid": "10160",
   "name": "Zabbix server"
   }
   ],
   "id": 1
}
```

Получить хосты с проблемами уровня ”средние” (Average) и ”высокие” (High).

Запрос:

```
{
   "jsonrpc": "2.0",
   "method": "host.get",
   "params": {
   "output": ["name"],
   "severities": [3, 4]
   },
   "auth": "038e1d7b1735c6a5436ee9eae095879e",
   "id": 1
}
```

Ответ:

```
{
   "jsonrpc": "2.0",
   "result": [
   {
   "hostid": "20170",
   "name": "Database"
   },
   {
   "hostid": "20183",
   "name": "workstation"
   }
   ],
   "id": 1
}
```


Получение элементов данных узла сети

Получите элементы данных для узла сети "Zabbix server". Запрос ограничен свойствами элемента данных `itemid`, `name` и `status`. Из-за большого размера ответа в примере показана только часть элементов данных.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```


Ответ:


```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "        ,                Zabbix",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "       Zabbix",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "               Zabbix",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "                                ",
          "status": "0"
        },
```

```
        {
          "itemid": "42252",
          "name": "                    ",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "      enp0s3:               ",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "      enp0s3:                ",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "      enp0s3:          ",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "      enp0s3:                ",
          "status": "0"
        },
        {
          "itemid": "69873",
          "name": "      enp0s3:                 ",
          "status": "0"
        },
        {
          "itemid": "69874",
          "name": "      enp0s3:           ",
          "status": "0"
        },
        {
          "itemid": "42253",
          "name": "           ",
          "status": "0"
        },
        {
          "itemid": "42254",
          "name": "                  ",
          "status": "0"
        },
        {
          "itemid": "42255",
          "name": "             ",
          "status": "0"
        }
      ]
    }
  ],
  "id": 1
}
```


См. также

- Группа узлов сети
- Шаблон
- Пользовательский макрос
- Интерфейс узла сети
- Прокси

- Группа прокси

Источник

CHost::get() в *ui/include/classes/api/services/CHost.php*.

host.massadd

Описание

`object host.massadd(object parameters)`

Этот метод позволяет одновременно добавлять несколько связанных объектов ко всем указанным узлам сети.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object)` Параметры, содержащие IDs узлов сети, которые нужно обновить, и объекты, которые нужно добавить ко всем узлам сети.

Метод принимает следующие параметры.

Parameter	Type	Description
hosts	object/array	Узлы сети, которые нужно обновить. У узлов сети должно быть определено только свойство `hostid`. Поведение параметра: - *required*
groups	object/array	Группы узлов сети, которые нужно добавить к указанным узлам сети. У групп узлов сети должно быть определено только свойство `groupid`.
interfaces	object/array	Интерфейсы узлов сети, которые нужно создать для указанных узлов сети.
macros	object/array	Пользовательские макросы, которые нужно создать для указанных узлов сети.
templates	object/array	Шаблоны, которые нужно привязать к указанным узлам сети. У шаблонов должно быть определено только свойство `templateid`.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных хостов. под свойством `hostids`.

Примеры

Добавление макросов

Добавьте два новых макроса к двум узлам сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
```

```
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Test description"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

См. также

- host.update
- Группа узлов сети
- Шаблон
- Пользовательский макрос
- Интерфейс узла сети

Источник

CHost::massAdd() в *ui/include/classes/api/services/CHost.php*.

host.massremove

Описание

`object host.massremove(object parameters)`

Этот метод позволяет удалить связанные объекты у нескольких узлов сети.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие IDs узлов сети, которые нужно обновить, и объекты, которые следует удалить.

Parameter	Type	Description
hostids	ID/array	IDs узлов сети, которые нужно обновить. Поведение параметра: - *required*

Parameter	Type	Description
groupids	ID/array	IDs групп узлов сети, из которых нужно удалить указанные узлы сети.
interfaces	object/array	Интерфейсы узла сети, которые нужно удалить у указанных узлов сети. Объект интерфейса узла сети должен содержать только свойства `ip`, `dns` и `port`.
macros	string/array	Пользовательские макросы, которые нужно удалить у указанных узлов сети.
templateids	ID/array	IDs шаблонов, которые нужно отвязать от указанных узлов сети.
templateids_clear	ID/array	IDs шаблонов, которые нужно отвязать и очистить у указанных узлов сети.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных хостов. под свойством `hostids`.

Примеры

Отвязка шаблонов

Отвяжите шаблон от двух узлов сети и удалите все сущности, унаследованные от шаблона.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

См. также

- host.update
- Пользовательский макрос
- Интерфейс узла сети

Источник

CHost::massRemove() в *ui/include/classes/api/services/CHost.php*.

host.update

Описание

`object host.update(object/array hosts)`

Этот метод позволяет обновлять существующие узлы сети.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию см. в разделе User roles.

Параметры

`(object/array)` Свойства узла сети, которые необходимо обновить.

Свойство `hostid` должно быть определено для каждого узла сети, все остальные свойства являются необязательными. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Однако обратите внимание, что при обновлении технического имени узла сети также будет обновлено видимое имя узла сети (если оно не задано отдельно) на значение технического имени узла сети.

В дополнение к стандартным свойствам узла сети, метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы узлов сети, которые заменят текущие группы узлов сети, к которым принадлежит узел сети. Все группы узлов сети, не указанные в запросе, будут отвязаны. Группы узлов сети должны содержать только свойство `groupid`.
interfaces	object/array	Интерфейсы узлов сети, которые заменят текущие интерфейсы узла сети. Все интерфейсы, не указанные в запросе, будут удалены.
tags	object/array	Теги узла сети, которые заменят текущие теги узла сети. Все теги, не указанные в запросе, будут удалены.
inventory	object	Свойства инвентаря узла сети.
macros	object/array	Пользовательские макросы, которые заменят текущие пользовательские макросы. Все макросы, не указанные в запросе, будут удалены.
templates	object/array	Шаблоны, которые заменят текущие привязанные шаблоны. Все шаблоны, не указанные в запросе, будут только отвязаны. Шаблоны должны содержать только свойство `templateid`.
templates_clear	object/array	Шаблоны, которые нужно отвязать и очистить у узла сети. Шаблоны должны содержать только свойство `templateid`.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных хостов. под свойством `hostids`.

Примеры

Активация узла сети

Активируйте мониторинг узла сети, установив для него статус "0".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Включение нескольких узлов сети

Включите мониторинг двух узлов сети, то есть установите их статус в ”0”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

Отвязывание шаблонов

Отвяжите и очистите два шаблона от узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Обновление макросов узла сети

Замените все макросы узла сети двумя новыми.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Updated description"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Обновление инвентаря узла сети

Изменение режима инвентаря и добавление местоположения

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Обновление тегов узла сети

Замените все теги узла сети новым тегом.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Обновление обнаруженных макросов узла сети

Преобразование созданного правилом обнаружения ”автоматического” макроса в ”ручной” и изменение его значения на ”new-value”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Обновление шифрования узла сети

Обновите узел сети "10590", чтобы использовать шифрование PSK только для соединений от узла сети к серверу Zabbix, и измените идентификатор PSK и ключ PSK. Обратите внимание, что узел сети должен быть предварительно настроен для использования PSK.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10590",
        "tls_connect": 1,
        "tls_accept": 2,
        "tls_psk_identity": "PSK 002",
        "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

См. также

- host.massadd
- host.massremove
- Группа узлов сети
- Шаблон
- Пользовательский макрос
- Интерфейс узла сети
- Инвентаризация узла сети
- Тег узла сети
- Прокси
- Группа прокси

Источник

CHost::update() в *ui/include/classes/api/services/CHost.php*.

Устройство

Этот класс предназначен для работы с мобильными устройствами, используемыми с мобильным приложением Zabbix.

Ссылки на объекты:

- Устройство

Доступные методы:

- device.get - получение устройств

- device.offboard - удаление устройств (вывод из эксплуатации)

Объект устройства

Объект устройства представляет мобильное устройство, используемое с мобильным приложением Zabbix.

Property	Type	Description
deviceid	ID	Идентификатор устройства (внутренний, например, 1, 2). Property behavior: - Только для чтения
userid	ID	Идентификатор пользователя устройства. Property behavior: - Постоянный
uuid	string	Универсальный уникальный идентификатор устройства (UUIDv7) (внешний, например, `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Property behavior: - Только для чтения
name	string	Имя устройства. Property behavior: - Только для чтения
status	integer	Состояние устройства. Возможные значения: 0 - Новое 1 - Активное 2 - Неназначенное Property behavior: - Только для чтения
activated_at	timestamp	Дата и время добавления устройства в мобильное приложение или 0, если оно еще не добавлено. Property behavior: - Только для чтения
lastaccess	timestamp	Дата и время последней аутентификации устройства в мобильном приложении или 0, если аутентификация еще не выполнялась. Property behavior: - Только для чтения

device.get

Этот метод позволяет получить мобильные устройства для использования с мобильным приложением Zabbix.

Метод доступен пользователям с ролями *User*, *Admin* и *Super admin*, если включен параметр `EnableMobileDevices` и у пользователя есть доступ к устройствам. Доступ к методам API можно отозвать в настройках роли пользователя.

Пользователи с ролями *User* и *Admin* могут получать только собственные устройства.

Сигнатура метода: `integer/array device.get(object parameters)`

Исходный код метода: `CDevice::get()` в `ui/include/classes/api/services/CDevice.php`

Параметры

Параметр	Тип	Описание
deviceids	ID/array	Возвращать только устройства с указанными идентификаторами устройств.
userids	ID/array	Возвращать только устройства, принадлежащие пользователям с указанными идентификаторами пользователей.
roleids	ID/array	Возвращать только устройства, принадлежащие пользователям с указанными идентификаторами ролей пользователей.
usrgrpids	ID/array	Возвращать только устройства, принадлежащие пользователям с указанными идентификаторами групп пользователей.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `deviceid`, `name`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

Возвращает либо массив объектов, либо количество полученных объектов (если используется `countOutput`).

Примеры

Получение устройств

Получение устройств, принадлежащих пользователям со следующими идентификаторами: 1, 2, 3.
 Обратите внимание, что у пользователя 2 (гостя) нет устройств, а устройство 3 еще не было добавлено или аутентифицировано в мобильном приложении Zabbix.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
```

```
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```

device.offboard

Этот метод позволяет удалить (вывести из эксплуатации) мобильные устройства, используемые с мобильным приложением Zabbix.

Метод доступен пользователям с ролями *User*, *Admin* и *Super admin*, если включен параметр `EnableMobileDevices` и пользователь имеет доступ к устройствам. Доступ к методам API можно отозвать в настройках роли пользователя.

Пользователи с ролями *User* и *Admin* могут удалять только собственные устройства.

Сигнатура метода: `object device.offboard(object parameters)`

Исходный код метода: `CDevice::offboard()` в `ui/include/classes/api/services/CDevice.php`

Параметры

Параметр	Тип	Описание
uuid	string	Универсальный уникальный идентификатор устройства (UUIDv7). Соответствует `deviceid`.
force	boolean	Установите значение `true`, чтобы игнорировать ошибку «Устройство не найдено» и принудительно удалить устройство. По умолчанию: `false`

Возвращаемые значения

Возвращает объект, содержащий UUID удалённого устройства.

Примеры

Удаление устройства

Удаляет устройство с указанным UUID, игнорируя ошибку «Устройство не найдено».

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

Шаблон

Этот класс предназначен для работы с шаблонами.

Ссылки на объекты:

- Шаблон
- Тег шаблона

Доступные методы:

- template.create - создать новые шаблоны
- template.delete - удалить шаблоны
- template.get - получить шаблоны
- template.massadd - добавить связанные объекты к шаблонам
- template.massremove - удалить связанные объекты из шаблонов
- template.update - обновить шаблоны

Объект шаблона

Следующие объекты напрямую связаны с API `template`.

Шаблон

Объект шаблона имеет следующие свойства.

Property	Type	Description
templateid	ID	ID шаблона. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
description	text	Описание шаблона.
host	string	Техническое имя шаблона. Поведение свойства: - *обязательно* для операций создания
name	string	Отображаемое имя шаблона. По умолчанию: значение свойства `host`.
readme	text	Инструкции по настройке, специфичные для шаблона, для отображения в Мастере узлов сети. Поддерживает форматирование Markdown.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных шаблонов с уже существующими. Генерируется автоматически, если не указан.
vendor_name	string	Имя поставщика шаблона. Для операций создания оба значения `vendor_name` и `vendor_version` должны быть либо заданы, либо оставлены пустыми. Для операций обновления `vendor_version` можно оставить пустым, если в базе данных оно уже имеет значение.

Property	Type	Description
vendor_version	string	Версия поставщика шаблона. Для операций создания оба значения `vendor_name` и `vendor_version` должны быть либо заданы, либо оставлены пустыми. Для операций обновления `vendor_name` можно оставить пустым, если в базе данных оно уже имеет значение.
wizard_ready	integer	Доступен ли шаблон для выбора в Мастере узлов сети. Возможные значения: 0 - *(по умолчанию)* Недоступен. 1 - Доступен.

Тег шаблона

Объект тега шаблона имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега шаблона. Property behavior: - *required*
value	string	Значение тега шаблона. Property behavior: - *read-only*.
object	integer	Тип объекта, от которого был унаследован тег. Possible values: 0 - template. Property behavior: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Property behavior: - *read-only*.

template.create

Описание

`object template.create(object/array templates)`

Этот метод позволяет создавать новые шаблоны.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Шаблоны для создания.

В дополнение к стандартным свойствам шаблона, метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы шаблонов, к которым нужно добавить шаблон. Группы шаблонов должны содержать только свойство `groupid`. Поведение параметра: - *required*
tags	object/array	Теги шаблона.
templates	object/array	Шаблоны, которые нужно привязать к шаблону. Шаблоны должны содержать только свойство `templateid`.
macros	object/array	Пользовательские макросы, которые нужно создать для шаблона.

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных шаблонов, указанных в свойстве `templateids`. Порядок возвращаемых идентификаторов соответствует порядку переданных шаблонов.

Примеры

Создание шаблона

Создайте шаблон с тегами и свяжите с ним два шаблона.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "template.create",
    "params": {
        "host": "Linux template",
        "groups": {
            "groupid": 1
        },
        "templates": [
            {
                "templateid": "11115"
            },
            {
                "templateid": "11116"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Источник

CTemplate::create() в *ui/include/classes/api/services/CTemplate.php*.

template.delete

Описание

`object template.delete(array templateIds)`

Этот метод позволяет удалять шаблоны.

При удалении шаблона также удаляются все его сущности (элементы данных, триггеры, графики и т. д.) с любых узлов сети или шаблонов, к которым он привязан. Чтобы сохранить эти сущности, сначала отвяжите шаблон от соответствующих узлов сети или шаблонов с помощью методов `host.update`, `host.massremove`, `template.update` или `template.massremove`.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительные сведения см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых шаблонов.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удалённых шаблонов, указанных в свойстве `templateids`.

Примеры

Удаление нескольких шаблонов

Удаление двух шаблонов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Источник

CTemplate::delete() в *ui/include/classes/api/services/CTemplate.php*.

template.get

Описание

`integer/array template.get(object parameters)`

Этот метод позволяет получать шаблоны в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
templateids	ID/array	Возвращать только шаблоны с указанными ID шаблонов.
groupids	ID/array	Возвращать только шаблоны, которые принадлежат указанным группам шаблонов.
parentTemplateids	ID/array	Возвращать только шаблоны, к которым привязан указанный шаблон.
hostids	ID/array	Возвращать только шаблоны, которые привязаны к указанным узлам сети/шаблонам.
graphids	ID/array	Возвращать только шаблоны, которые содержат указанные графики.
itemids	ID/array	Возвращать только шаблоны, которые содержат указанные элементы данных.
triggerids	ID/array	Возвращать только шаблоны, которые содержат указанные триггеры.
with_items	flag	Возвращать только шаблоны, у которых есть элементы данных.
with_triggers	flag	Возвращать только шаблоны, у которых есть триггеры.
with_graphs	flag	Возвращать только шаблоны, у которых есть графики.
with_httptests	flag	Возвращать только шаблоны, у которых есть веб-сценарии.
evaltype	integer	Метод оценки тегов. Возможные значения: 0 - *(по умолчанию)* И/ИЛИ; 2 - ИЛИ.
tags	object/array	Возвращать только шаблоны с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все шаблоны. Возможные значения оператора: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
inheritedTags	boolean	Возвращать шаблоны, которые имеют указанные `tags` также в связанных шаблонах. Возможные значения: `true` - связанные шаблоны также должны иметь указанные теги; `false` - *(по умолчанию)* теги из связанных шаблонов игнорируются.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые находятся в связанных шаблонах.
selectTags	query	Возвращать теги шаблона в свойстве `tags`.
selectDiscoveryRules	query	Возвращать свойство `discoveryRules` с правилами LLD шаблона. Поддерживает `count`.
selectHosts	query	Возвращать узлы сети, привязанные к шаблону, в свойстве `hosts`. Поддерживает `count`.
selectTemplateGroups	query	Возвращать группы шаблонов, к которым принадлежит шаблон, в свойстве `templategroups`.

Parameter	Type	Description
selectTemplates	query	Возвращать шаблоны, к которым привязан указанный шаблон, в свойстве `templates`. Поддерживает `count`.
selectParentTemplates	query	Возвращать шаблоны, которые привязаны к указанному шаблону, в свойстве `parentTemplates`. Поддерживает `count`.
selectHttpTests	query	Возвращать веб-сценарии из шаблона в свойстве `httpTests`. Поддерживает `count`.
selectItems	query	Возвращать элементы данных из шаблона в свойстве `items`. Поддерживает `count`.
selectTriggers	query	Возвращать триггеры из шаблона в свойстве `triggers`. Поддерживает `count`.
selectGraphs	query	Возвращать графики из шаблона в свойстве `graphs`. Поддерживает `count`.
selectMacros	query	Возвращать макросы из шаблона в свойстве `macros`.
selectDashboards	query	Возвращать дашборды из шаблона в свойстве `dashboards`. Поддерживает `count`.
selectValueMaps	query	Возвращать свойство `valuemaps` с картами значений шаблона.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectTemplates` - результаты будут отсортированы по `name`; `selectHosts` - по `host`; `selectParentTemplates` - по `host`; `selectItems` - по `name`; `selectDiscoveryRules` - по `name`; `selectTriggers` - по `description`; `selectGraphs` - по `name`; `selectDashboards` - по `name`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Возвращать низкоуровневые обнаружения из шаблона в свойстве `discoveries`. Поддерживает `count`. Этот запрос **устарел**, вместо него используйте `selectDiscoveryRules`.

Возвращаемые значения

(integer/array) Возвращает либо:

- массив объектов;
- количество полученных объектов, если использован параметр countOutput.

Примеры

Получение шаблонов по имени

Получите все данные о двух шаблонах с именами "Linux by Zabbix agent" и "Windows by Zabbix agent".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": "                Linux.                Zabbix 8.0       .\r\n\r\n
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "##    \r\n\r\n                Linux.                Zabbix 8.0       .\r\n\r\n####
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
```

```
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": "                Windows.                Zabbix 8.0      .\r\n\r\n
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "##    \r\n\r\n                Windows.                Zabbix 8.0      ."
        }
    ],
    "id": 1
}
```


Получение групп шаблонов

Получение групп шаблонов, членом которых является шаблон "Linux by Zabbix agent".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
```

```
                }
            ]
        }
    ],
    "id": 1
}
```

Получение узлов сети по шаблону

Получите узлы сети, к которым привязан шаблон ”10001” (*Linux by Zabbix agent*).

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Поиск по тегам шаблона

Получить шаблоны, у которых тег ”host-name” равен ”{HOST.NAME}”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
```

```
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Группа шаблонов
- Шаблон
- Пользовательский макрос
- Интерфейс узла сети

Источник

CTemplate::get() в *ui/include/classes/api/services/CTemplate.php*.

template.massadd

Описание

`object template.massadd(object parameters)`

Этот метод позволяет одновременно добавлять несколько связанных объектов к указанным шаблонам.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. Роли пользователей.

Параметры

`(object)` Параметры, содержащие IDs шаблонов для обновления и объекты, которые нужно добавить в шаблоны.

Метод принимает следующие параметры.

Parameter	Type	Description
templates	object/array	Шаблоны, которые нужно обновить. Шаблоны должны содержать только свойство `templateid`. Поведение параметра: - *required*
groups	object/array	Группы шаблонов, в которые нужно добавить указанные шаблоны. Группы шаблонов должны содержать только свойство `groupid`.
macros	object/array	Пользовательские макросы, которые нужно создать для указанных шаблонов.

Parameter	Type	Description
templates_link	object/array	Шаблоны, которые нужно связать с указанными шаблонами. Шаблоны должны содержать только свойство `templateid`.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных шаблонов, указанных в свойстве `templateids`.

Примеры

Привязать группу к шаблонам

Добавить группу шаблонов ”2” к двум шаблонам.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Связать два шаблона с шаблоном

Свяжите шаблоны ”10106” и ”10104” с шаблоном ”10073”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
```

```
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

См. также

- template.update
- Узел сети
- Группа шаблонов
- Пользовательский макрос

Источник

CTemplate::massAdd() в *ui/include/classes/api/services/CTemplate.php*.

template.massremove

Описание

`object template.massremove(object parameters)`

Этот метод позволяет удалять связанные объекты из нескольких шаблонов.

> **Note:**
> Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, содержащие ID шаблонов, которые нужно обновить, и объекты, которые следует удалить.

Parameter	Type	Description
templateids	ID/array	ID шаблонов, которые нужно обновить. Поведение параметра: - *required*
groupids	ID/array	ID групп шаблонов, из которых нужно удалить указанные шаблоны.
macros	string/array	ID пользовательских макросов, которые нужно удалить из указанных шаблонов.
templateids_clear	ID/array	ID шаблонов, которые нужно отвязать и очистить от указанных шаблонов (upstream).
templateids_link	ID/array	ID шаблонов, которые нужно отвязать от указанных шаблонов (upstream).

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных шаблонов, указанных в свойстве `templateids`.

Примеры

Удаление шаблонов из группы

Удалите два шаблона из группы "2".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Отвязка шаблонов от узла сети

Отвязать шаблоны "10106" и "10104" от шаблона "10085".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

См. также

- template.update

- Пользовательский макрос

Источник

CTemplate::massRemove() в *ui/include/classes/api/services/CTemplate.php*.

template.update

Описание

`object template.update(object/array templates)`

Этот метод позволяет обновлять существующие шаблоны.

> **Note:**
> Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства шаблона, которые необходимо обновить.

Для каждого шаблона должно быть определено свойство `templateid`, все остальные свойства необязательны. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Помимо стандартных свойств шаблона, метод принимает следующие параметры.

Parameter	Type	Description
groups	object/array	Группы шаблонов, которыми нужно заменить текущие группы шаблонов, к которым принадлежат шаблоны. Группы шаблонов должны содержать только свойство `groupid`.
tags	object/array	Теги шаблона, которыми нужно заменить текущие теги шаблона.
macros	object/array	Пользовательские макросы, которыми нужно заменить текущие пользовательские макросы на указанных шаблонах.
templates	object/array	Шаблоны, которыми нужно заменить текущие связанные шаблоны. Шаблоны, которые не переданы, будут только отвязаны. Шаблоны должны содержать только свойство `templateid`.
templates_clear	object/array	Шаблоны, которые нужно отвязать и очистить от указанных шаблонов. Шаблоны должны содержать только свойство `templateid`.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных шаблонов, указанных в свойстве `templateids`.

Примеры

Изменение стандартных свойств шаблона

Измените техническое имя шаблона на "Linux by Zabbix agent Custom", видимое имя на "My template" и обновите описание шаблона.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "host": "Linux by Zabbix agent Custom",
        "name": "My template",
        "description": "This is a custom Linux template."
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Обновление групп шаблона

Замените все группы шаблона для указанного шаблона на другую.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Обновление нескольких групп шаблонов

Замените все группы шаблонов для нескольких шаблонов на другие.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10086",
            "groups": [
                {
                    "groupid": "24"
                }
            ]
        },
        {
            "templateid": "10087",
            "groups": [
```

```
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        },
        {
            "templateid": "10088",
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Обновление тегов шаблона

Замените все теги шаблона на другой.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
```

```
    },
    "id": 1
}
```

Обновление макросов шаблона

Замените все макросы шаблона на другие.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Обновление нескольких макросов шаблона

Замените все пользовательские макросы указанным пользовательским макросом в нескольких шаблонах.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "  -  ,                              .                ,
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "  -  ,                              .                ,
                    }
                ]
            }
```

```
        ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Обновление связанных шаблонов шаблона

Отвяжите (без очистки) все шаблоны от указанного шаблона и свяжите с ним другой.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Очистка нескольких связанных шаблонов шаблона

Отсоедините и очистите указанный связанный шаблон из заданных шаблонов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
```

```
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Источник

CTemplate::update() в *ui/include/classes/api/services/CTemplate.php*.

Элемент данных

Этот класс предназначен для работы с элементами данных.

Ссылки на объекты:

- Элемент данных
  - HTTP header
  - HTTP query field
- Тег элемента данных
- Предобработка элемента данных

Доступные методы:

- item.create - создать новые элементы данных
- item.delete - удалить элементы данных
- item.get - получить элементы данных
- item.update - обновить элементы данных

Объект элемента данных

Следующие объекты напрямую связаны с API `item`.

Элемент данных

> **Note:**
> Веб-элементы нельзя напрямую создавать, обновлять или удалять через API Zabbix.

Объект элемента данных имеет следующие свойства.

Property	Type	Description
itemid	ID	Идентификатор элемента данных. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Property	Type	Description
delay	string	Интервал обновления элемента данных. Принимает количество секунд (например, 1, 60, 300), значение с суффиксом времени (например, 30s, 1m, 2h, 1d) или, опционально, один или несколько пользовательских интервалов, разделённых точками с запятой. Пользовательские интервалы могут представлять собой комбинацию гибких интервалов и интервалов расписания. Поддерживает пользовательские макросы. При использовании макроса значение должно состоять из одного макроса. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы можно записывать как два макроса, разделённых косой чертой (например, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Пример: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Поведение свойства: - *обязательно*, если type имеет значение ”Zabbix agent” (0), ”Simple check” (3), ”Zabbix internal” (5), ”External check” (10), ”Database monitor” (11), ”IPMI agent” (12), ”SSH agent” (13), ”TELNET agent” (14), ”Calculated” (15), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”Browser” (22), ”Telemetry query” (24), или если type имеет значение ”Zabbix agent (active)” (7), а key_ не содержит ”mqtt.get”
hostid	ID	Идентификатор узла сети или шаблона, которому принадлежит элемент данных. Поведение свойства: - *константа* - *обязательно* для операций создания
interfaceid	ID	Идентификатор интерфейса узла сети элемента данных. Поведение свойства: - *обязательно*, если элемент данных принадлежит узлу сети, а type имеет значение ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” или ”SNMP agent” - *поддерживается*, если элемент данных принадлежит узлу сети, а type имеет значение ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent” - *только для чтения* для обнаруженных объектов
key_	string	Ключ элемента данных. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных или обнаруженных объектов
name	string	Имя элемента данных. Поддерживает пользовательские макросы. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных или обнаруженных объектов
name_resolved	string	Имя элемента данных с разрешёнными пользовательскими макросами. Поведение свойства: - *только для чтения*

Property	Type	Description
type	integer	Тип элемента данных. Возможные значения: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных или обнаруженных объектов
url	string	Строка URL. Поддерживает пользовательские макросы, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
value_type	integer	Тип информации элемента данных. Возможные значения: 0 - числовой с плавающей запятой; 1 - символьный; 2 - журнал; 3 - числовой без знака; 4 - текстовый; 5 - двоичный; 6 - JSON. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных или обнаруженных объектов
allow_traps	integer	Разрешить заполнение значения аналогично элементу данных типа trapper. 0 - *(по умолчанию)* не принимать входящие данные; 1 - принимать входящие данные. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для обнаруженных объектов

Property	Type	Description
authtype	integer	Метод аутентификации. Возможные значения, если type имеет значение ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если type имеет значение ”HTTP agent”: 0 - *(по умолчанию)* отсутствует; 1 - базовая; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если type имеет значение ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если type имеет значение ”HTTP agent”) или обнаруженных объектов
description	string	Описание элемента данных. Поведение свойства: - *только для чтения* для обнаруженных объектов
error	string	Текст ошибки, если возникли проблемы при обновлении значения элемента данных. Поведение свойства: - *только для чтения*
flags	integer	Источник элемента данных. Возможные значения: 0 - обычный элемент данных; 4 - элемент данных, преобразованный из прототипа. Поведение свойства: - *только для чтения*
follow_redirects	integer	Следовать перенаправлениям ответа при опросе данных. Возможные значения: 0 - не следовать перенаправлениям; 1 - *(по умолчанию)* следовать перенаправлениям. Поведение свойства: - *поддерживается*, если type имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
headers	array	Массив заголовков, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если type имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
history	string	Единица времени, определяющая срок хранения данных истории. Также поддерживает пользовательский макрос. По умолчанию: 31d. Поведение свойства: - *только для чтения* для обнаруженных объектов

Property	Type	Description
http_proxy	string	Строка подключения прокси HTTP(S). Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
inventory_link	integer	Идентификатор поля инвентаризации узла сети, которое заполняется элементом данных. Список поддерживаемых полей инвентаризации узла сети и их идентификаторов приведён на странице инвентаризации узла сети. По умолчанию: 0. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float”, ”character”, ”numeric unsigned” или ”text” - *только для чтения* для обнаруженных объектов
ipmi_sensor	string	Датчик IPMI. Поведение свойства: - *обязательно*, если `type` имеет значение ”IPMI agent”, а `key_` не имеет значение ”ipmi.get” - *поддерживается*, если `type` имеет значение ”IPMI agent” - *только для чтения* для унаследованных или обнаруженных объектов
jmx_endpoint	string	Пользовательская строка подключения агента JMX. Значение по умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` имеет значение ”JMX agent” - *только для чтения* для обнаруженных объектов
lastclock	timestamp	Время последнего обновления значения элемента данных. По умолчанию отображаются только значения за последние 24 часа. Этот период можно увеличить, изменив значение параметра *Максимальный период отображения истории* в разделе меню *Администрирование > Общие*. Поведение свойства: - *только для чтения*
lastns	integer	Наносекунды на момент последнего обновления значения элемента данных. По умолчанию отображаются только значения за последние 24 часа. Этот период можно увеличить, изменив значение параметра *Максимальный период отображения истории* в разделе меню *Администрирование > Общие*. Поведение свойства: - *только для чтения*

Property	Type	Description
lastvalue	string	Последнее значение элемента данных. По умолчанию отображаются только значения за последние 24 часа. Этот период можно увеличить, изменив значение параметра *Максимальный период отображения истории* в разделе меню *Администрирование > Общие*. Поведение свойства: - *только для чтения*
logtimefmt	string	Формат времени в записях журнала. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”log” - *только для чтения* для унаследованных или обнаруженных объектов
master_itemid	ID	Идентификатор главного элемента данных. Поведение свойства: - *обязательно*, если `type` имеет значение ”Dependent item” - *только для чтения* для унаследованных или обнаруженных объектов
output_format	integer	Следует ли преобразовать ответ в JSON. 0 - *(по умолчанию)* сохранять без изменений; 1 - преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
params	string	Дополнительные параметры в зависимости от типа элемента данных: - выполняемый скрипт для элементов данных SSH agent и TELNET agent; - SQL-запрос для элементов данных Database monitor; - формула для вычисляемых элементов данных; - скрипт для элементов данных Script и Browser. Поведение свойства: - *обязательно*, если `type` имеет значение ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”Script” или ”Browser”) или обнаруженных объектов
parameters	object/array	Дополнительные параметры, если `type` имеет значение ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должен быть уникальным. Поведение свойства: - *поддерживается*, если `type` имеет значение ”Script” или ”Browser” - *только для чтения* для унаследованных или обнаруженных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”JMX agent”, а `username` задан - *поддерживается*, если `type` имеет значение ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”) или обнаруженных объектов

Property	Type	Description
post_type	integer	Тип тела данных POST, хранящегося в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* необработанные данные; 2 - данные JSON; 3 - данные XML. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
posts	string	Данные тела запроса HTTP(S). Поведение свойства: - *обязательно*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”JSON data” или ”XML data” - *поддерживается*, если `type` имеет значение ”HTTP agent”, а `post_type` имеет значение ”Raw data” - *только для чтения* для унаследованных или обнаруженных объектов
prevvalue	string	Предыдущее значение элемента данных. По умолчанию отображаются только значения за последние 24 часа. Этот период можно увеличить, изменив значение параметра *Максимальный период отображения истории* в разделе меню *Администрирование > Общие*. Поведение свойства: - *только для чтения*
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key” - *только для чтения* для обнаруженных объектов
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, а `authtype` имеет значение ”public key” - *только для чтения* для обнаруженных объектов
query_fields	array	Массив полей запроса, отправляемых при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
request_method	integer	Тип метода запроса. Возможные значения: 0 - *(по умолчанию)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов

Property	Type	Description
retrieve_mode	integer	Какая часть ответа должна сохраняться. Возможные значения, если `request_method` имеет значение ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* тело; 1 - заголовки; 2 - сохранять тело и заголовки. Возможные значения, если `request_method` имеет значение ”HEAD”: 1 - заголовки. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
snmp_oid	string	OID SNMP. Поведение свойства: - *обязательно*, если `type` имеет значение ”SNMP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_cert_file	string	Путь к файлу открытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_key_file	string	Путь к файлу закрытого ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_key_password	string	Пароль для файла ключа SSL. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
state	integer	Состояние элемента данных. Возможные значения: 0 - *(по умолчанию)* нормальное; 1 - не поддерживается. Поведение свойства: - *только для чтения*
status	integer	Состояние элемента данных. Возможные значения: 0 - *(по умолчанию)* элемент данных включён; 1 - элемент данных отключён.

Property	Type	Description
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделённые запятыми. Также поддерживает пользовательские макросы как часть списка, разделённого запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` имеет значение "HTTP agent" - *только для чтения* для унаследованных или обнаруженных объектов
templateid	ID	Идентификатор элемента данных родительского шаблона. *Подсказка*: Используйте свойство `hostid`, чтобы указать шаблон, которому принадлежит элемент данных. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса на опрос данных элемента данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, `300s`, `1m`). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-600s. По умолчанию: "" - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` имеет значение "Zabbix agent" (0), "Simple check" (3), а `key_` не начинается с "`vmware.`" и "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20), а `snmp_oid` начинается с "`walk[`" или "`get[`", "Script" (21), "Browser" (22) - *только для чтения* для унаследованных и обнаруженных объектов, - если применяется значение по умолчанию и элемент данных принадлежит узлу сети, который отслеживается недоступным прокси или группой прокси, используется глобальный тайм-аут.
trapper_hosts	string	Разрешённые узлы сети. Поведение свойства: - *только для чтения* для обнаруженных объектов - *поддерживается*, если `type` имеет значение "Zabbix trapper" или если `type` имеет значение "HTTP agent", а `allow_traps` имеет значение "Allow incoming data". Если свойство не указано, ему автоматически присваивается пользовательский макрос `{$TRAPPER.ALLOWED_HOSTS}` (значение по умолчанию). По умолчанию `{$TRAPPER.ALLOWED_HOSTS}` имеет значение `127.0.0.1,::1`. Если свойству присвоена пустая строка, входящие подключения от всех узлов сети будут запрещены.
trends	string	Единица времени, определяющая срок хранения данных трендов. Также поддерживает пользовательский макрос. По умолчанию: 365d. Поведение свойства: - *поддерживается*, если `value_type` имеет значение "numeric float" или "numeric unsigned" - *только для чтения* для обнаруженных объектов

Property	Type	Description
units	string	Единицы измерения значения. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float” или ”numeric unsigned” - *только для чтения* для унаследованных или обнаруженных объектов
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` имеет значение ”SSH agent”, ”TELNET agent” или если `type` имеет значение ”JMX agent”, а `password` задан - *поддерживается*, если `type` имеет значение ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` имеет значение ”HTTP agent”) или обнаруженных объектов
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированного элемента данных с уже существующими. Создаётся автоматически, если не указан. Поведение свойства: - *поддерживается*, если элемент данных принадлежит шаблону
valuemapid	ID	Идентификатор связанной карты значений. Поведение свойства: - *поддерживается*, если `value_type` имеет значение ”numeric float”, ”character” или ”numeric unsigned” - *только для чтения* для унаследованных или обнаруженных объектов
verify_host	integer	Проверять ли соответствие имени узла сети для подключения имени, указанному в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
verify_peer	integer	Проверять ли подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* не проверять; 1 - проверять. Поведение свойства: - *поддерживается*, если `type` имеет значение ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
query	object	Конфигурация запроса телеметрии. Определяет, какие данные элемент данных запрашивает из источника телеметрии, как фильтрует данные и как агрегирует результат. Поведение свойства: - *обязательно*, если `type` имеет значение ”Telemetry query” - *только для чтения* для унаследованных или обнаруженных объектов

Property	Type	Description
time_shift	string	Насколько далеко в прошлое (до текущего времени) заканчивается временной диапазон, который запрашивает элемент данных. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 0-86400s. По умолчанию: 15s. Поведение свойства: - *поддерживается*, если type имеет значение "Telemetry query"
lookback_limit	string	Насколько далеко в прошлое (до текущего времени) элемент данных запрашивает данные, если отсутствует предыдущий запрос, который можно продолжить (например, при первом запуске элемента данных), или если предыдущий запрос старше delay (например, из-за недоступности сервера Zabbix). Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-259200s. По умолчанию: 10m. Поведение свойства: - *поддерживается*, если type имеет значение "Telemetry query"
granularity	string	Длительность части, на которую разбивается полный временной диапазон запроса. Принимает количество секунд (например, 1, 60, 300) или значение с суффиксом времени (например, 30s, 1m, 2h, 1d). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-86400s. По умолчанию: 15s. Поведение свойства: - *поддерживается*, если type имеет значение "Telemetry query"

HTTP-заголовок

Объект заголовка имеет следующие свойства:

Property	Type	Description
name	string	Имя HTTP-заголовка. Поведение свойства: - *required*
value	string	Значение заголовка. Поведение свойства: - *required*

Поле HTTP-запроса

Объект поля запроса определяет имя и значение, которые используются для указания параметра URL. Он имеет следующие свойства:

Property	Type	Description
name	string	Имя параметра. Property behavior: - *required*
value	string	Значение параметра. Property behavior: - *required*

Запрос телеметрии

Объект запроса телеметрии имеет следующие свойства.

Property	Type	Description
signal_type	integer	Тип сигнала, который запрашивает элемент данных. Возможные значения: 0 - Трассировки; 1 - Метрики; 2 - Журналы. Поведение свойства: - *обязательное*
metric_point_type	integer	Тип точки метрики, который запрашивает элемент данных. Возможные значения: 0 - Сумма; 1 - Показатель; 2 - Гистограмма; 3 - Экспоненциальная гистограмма. Поведение свойства: - *обязательное*, если для `signal_type` установлено значение ”Metrics”
columns	array	Массив столбцов данных телеметрии, которые запрашивает элемент данных. Может быть пустым. Поведение свойства: - *обязательное*
aggregated_columns	array	Массив агрегированных столбцов данных телеметрии, которые запрашивает элемент данных, а также способ вычисления значения на основе записей столбцов. Поведение свойства: - *обязательное*
filter	object	Фильтр запроса телеметрии для записей столбцов в запросе. Поведение свойства: - *обязательное*

Столбцы данных телеметрии

Объект столбцов данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, который запрашивает элемент данных. См. поддерживаемые столбцы данных телеметрии. Для нескольких столбцов с одиночными значениями каждое значение `column` должно быть уникальным. Для нескольких столбцов, содержащих пары «ключ-значение», каждая пара `column` и `attribute_key` должна быть уникальной.
attribute_key	string	Имя ключа, который запрашивает элемент данных, в столбце, содержащем пары «ключ-значение». Для нескольких столбцов, содержащих пары «ключ-значение», каждая пара `column` и `attribute_key` должна быть уникальной. Поведение свойства: - *обязательно*, если `column` задано как столбец, содержащий пары «ключ-значение»

Агрегированные столбцы данных телеметрии

Объект агрегированных столбцов данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, который запрашивает элемент данных. См. поддерживаемые столбцы данных телеметрии. Для нескольких столбцов каждое значение `column` должно быть уникальным. Это свойство не поддерживается, если для `function` установлено значение ”Count”. Поведение свойства: - *обязательно*, если для `function` установлено значение ”Min”, ”Max”, ”Average”, ”Sum” или ”Percentile”
function	integer	Функция, применяемая элементом данных к записям столбца. Возможные значения: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Поведение свойства: - *обязательно*
parameters	array	Процентиль, который элемент данных вычисляет по записям столбца. Возможное значение: число от 0 до 100, заданное в виде строки. Поведение свойства: - *обязательно*, если для `function` установлено значение ”Percentile”

Property	Type	Description
alias	string	Уникальное имя вычисленного значения, используемое для его идентификации в значении элемента данных. Каждое значение `alias` должно быть уникальным. Поведение свойства: - *обязательно*

Фильтр данных телеметрии

Объект фильтра данных телеметрии имеет следующие свойства.

Property	Type	Description
conditions	array	Условия фильтра данных телеметрии, которым должна соответствовать запись в столбце данных телеметрии. Поведение свойства: - *обязательное*
evaltype	integer	Метод вычисления условий фильтра. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 3 - Пользовательское выражение. Поведение свойства: - *обязательное*
eval_formula	string	Сгенерированное выражение, используемое для вычисления условий фильтра. Выражение содержит идентификаторы, ссылающиеся на определенные условия фильтра по их `formulaid`. Значение `eval_formula` равно значению `formula` для фильтров с пользовательским выражением. Поведение свойства: - *только для чтения*
formula	string	Определенное пользователем выражение, используемое для вычисления условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, ссылающиеся на определенные условия фильтра по их `formulaid`. Идентификаторы, используемые в выражении, должны в точности соответствовать идентификаторам, определенным в условиях фильтра: ни одно условие не должно оставаться неиспользованным или быть пропущенным. Поведение свойства: - *обязательное*, если для `evaltype` установлено значение "Пользовательское выражение"

Условия фильтрации данных телеметрии

Объект условий фильтрации данных телеметрии имеет следующие свойства.

Property	Type	Description
column	string	Имя столбца данных телеметрии, к которому применяется условие фильтрации. См. поддерживаемые столбцы данных телеметрии. Поведение свойства: - *обязательное*
attribute_key	string	Имя ключа, к которому применяется условие, в столбце, содержащем пары «ключ-значение». Поведение свойства: - *обязательное*, если `column` задан как столбец, содержащий пары «ключ-значение»
operator	integer	Оператор условия фильтрации. Возможные значения для столбцов с одиночными значениями: 0 - Равно; 1 - Не равно; 2 - Содержит; 3 - Не содержит. Возможные значения для столбцов, содержащих пары «ключ-значение»: 0 - Равно; 1 - Не равно; 12 - Существует. Поведение свойства: - *обязательное*
value	string	Значение для сравнения. Поведение свойства: - *обязательное*, если `operator` задан как ”Равно”, ”Не равно”, ”Содержит” или ”Не содержит”
formulaid	string	Произвольный уникальный идентификатор, используемый для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. Идентификатор должен быть задан пользователем при изменении условий фильтрации, но при последующем запросе будет сгенерирован заново. Поведение свойства: - *обязательное*, если `evaltype` объекта Фильтр запроса телеметрии задан как ”Пользовательское выражение”

Тег элемента данных

Объект тега элемента данных имеет следующие свойства.

Property	Type	Description
tag	string	Имя тега элемента данных. Поведение свойства: - *required*
value	string	Значение тега элемента данных. Поведение свойства: - *read-only*.

Property	Type	Description
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети. Поведение свойства: - *read-only*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *read-only*.

Предобработка элемента данных

Объект предобработки элемента данных имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип параметра предобработки. Возможные значения: 1 - Пользовательский множитель; 2 - Удаление символов справа; 3 - Удаление символов слева; 4 - Удаление символов с обеих сторон; 5 - Регулярное выражение; 6 - Преобразование логического значения в десятичное число; 7 - Преобразование восьмеричного числа в десятичное; 8 - Преобразование шестнадцатеричного числа в десятичное; 9 - Простое изменение; 10 - Изменение в секунду; 11 - XML XPath; 12 - JSONPath; 13 - В диапазоне; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка наличия ошибки в JSON; 17 - Проверка наличия ошибки в XML; 18 - Проверка наличия ошибки с помощью регулярного выражения; 19 - Отбрасывать неизменившиеся значения; 20 - Отбрасывать неизменившиеся значения с контролем доступности; 21 - JavaScript; 22 - Шаблон Prometheus; 23 - Преобразование Prometheus в JSON; 24 - Преобразование CSV в JSON; 25 - Замена; 26 - Проверка неподдерживаемого значения; 27 - Преобразование XML в JSON; 28 - Значение SNMP walk; 29 - Преобразование SNMP walk в JSON; 30 - Значение SNMP get. Поведение свойства: - *обязательное*

Свойство	Тип	Описание
params	string	Дополнительные параметры, используемые параметром предобработки. Несколько параметров разделяются символом новой строки (\n). Если для `type` установлено значение «Проверка неподдерживаемого значения», параметры соответствуют синтаксису `<scope>[\n<pattern>]`, где *pattern* - регулярное выражение, а *scope* имеет одно из следующих значений: -1 - соответствует любой ошибке; 0 - проверять, соответствует ли сообщение об ошибке шаблону *pattern*; 1 - проверять, не соответствует ли сообщение об ошибке шаблону *pattern*. Поведение свойства: - *обязательное*, если для `type` установлено значение «Пользовательский множитель» (1), «Удаление символов справа» (2), «Удаление символов слева» (3), «Удаление символов с обеих сторон» (4), «Регулярное выражение» (5), «XML XPath» (11), «JSONPath» (12), «В диапазоне» (13), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Проверка наличия ошибки с помощью регулярного выражения» (18), «Отбрасывать неизменившиеся значения с контролем доступности» (20), «JavaScript» (21), «Шаблон Prometheus» (22), «Преобразование Prometheus в JSON» (23), «Преобразование CSV в JSON» (24), «Замена» (25), Проверка неподдерживаемого значения (26), «Значение SNMP walk» (28), «Преобразование SNMP walk в JSON» (29) или «Значение SNMP get» (30)
error_handler	integer	Тип действия, используемый в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке задается сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Возможные значения, если для `type` установлено значение «Проверка неподдерживаемого значения»: 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательное*, если для `type` установлено значение «Пользовательский множитель» (1), «Регулярное выражение» (5), «Преобразование логического значения в десятичное число» (6), «Преобразование восьмеричного числа в десятичное» (7), «Преобразование шестнадцатеричного числа в десятичное» (8), «Простое изменение» (9), «Изменение в секунду» (10), «XML XPath» (11), «JSONPath» (12), «В диапазоне» (13), «Соответствует регулярному выражению» (14), «Не соответствует регулярному выражению» (15), «Проверка наличия ошибки в JSON» (16), «Проверка наличия ошибки в XML» (17), «Проверка наличия ошибки с помощью регулярного выражения» (18), «Шаблон Prometheus» (22), «Преобразование Prometheus в JSON» (23), «Преобразование CSV в JSON» (24), «Проверка неподдерживаемого значения» (26), «Преобразование XML в JSON» (27), «Значение SNMP walk» (28), «Преобразование SNMP walk в JSON» (29) или «Значение SNMP get» (30)

Свойство	Тип	Описание
error_handler_params	string	Параметры обработчика ошибок. Поведение свойства: - *обязательное*, если для `error_handler` установлено значение «Установить пользовательское значение» или «Установить пользовательское сообщение об ошибке»

Для каждого типа предобработки поддерживаются следующие параметры и обработчики ошибок.

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
1	Пользовательский множитель	множитель^1,6^			0, 1, 2, 3
2	Удаление символов справа	список символов[2]			
3	Удаление символов слева	список символов[2]			
4	Удаление символов с обеих сторон	список символов[2]			
5	Регулярное выражение	шаблон[3]	вывод[2]		0, 1, 2, 3
6	Преобразование логического значения в десятичное число				0, 1, 2, 3
7	Преобразование восьмеричного числа в десятичное				0, 1, 2, 3
8	Преобразование шестнадцатеричного числа в десятичное				0, 1, 2, 3
9	Простое изменение				0, 1, 2, 3
10	Изменение в секунду				0, 1, 2, 3
11	XML XPath	путь[4]			0, 1, 2, 3
12	JSONPath	путь[4]			0, 1, 2, 3
13	В диапазоне	минимум^1,6^	максимум^1,6^		0, 1, 2, 3
14	Соответствует регулярному выражению	шаблон[3]			0, 1, 2, 3
15	Не соответствует регулярному выражению	шаблон[3]			0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
16	Проверка наличия ошибки в JSON	путь[4]			0, 1, 2, 3
17	Проверка наличия ошибки в XML	путь[4]			0, 1, 2, 3
18	Проверка наличия ошибки с помощью регулярного выражения	шаблон[3]	вывод[2]		0, 1, 2, 3
19	Отбрасывать неизменившиеся значения				
20	Отбрасывать неизменившиеся значения с контролем доступности	секунды^5, 6^			
21	JavaScript	скрипт[2]			
22	Шаблон Prometheus	шаблон^6, 7^	`value`, `label`, `function`	вывод^8, 9^	0, 1, 2, 3
23	Преобразование Prometheus в JSON	шаблон^6, 7^			0, 1, 2, 3
24	Преобразование CSV в JSON	символ-разделитель[2]	символ[2]	0,1	0, 1, 2, 3
25	Замена	искомая строка[2]	замена[2]		
26	Проверка неподдерживаемого значения	область[1]	шаблон^3, 6^		1, 2, 3
27	Преобразование XML в JSON				0, 1, 2, 3
28	Значение SNMP walk	OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS		0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
29	Преобразование SNMP walk в JSON[10]	Имя поля[2]	Префикс OID[2]	Формат: 0 - Без изменений 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS	0, 1, 2, 3
30	Значение SNMP get	Формат: 1 - UTF-8 из Hex-STRING 2 - MAC из Hex-STRING 3 - Целое число из BITS			0, 1, 2, 3

[1] число с плавающей точкой (целые числа неявно преобразуются в значения с плавающей точкой)
 [2] строка
 [3] регулярное выражение
 [4] JSONPath или XML XPath
 [5] положительное целое число (с поддержкой суффиксов времени, например 30s, 1m, 2h, 1d)
 [6] пользовательский макрос
 [7] шаблон Prometheus, соответствующий синтаксису: `<metric name>{<label name>="<label value>", ...} == <value>`. Каждый компонент шаблона Prometheus (метрика, имя метки, значение метки и значение метрики) может быть пользовательским макросом.
 [8] вывод Prometheus, соответствующий синтаксису: `<label name>` (может быть пользовательским макросом), если в качестве второго параметра выбран `label`.
 [9] Одна из функций агрегации: `sum`, `min`, `max`, `avg`, `count`, если в качестве второго параметра выбрано `function`.
 [10] Поддерживает несколько записей «Имя поля,Префикс OID,Формат», разделенных символом новой строки.

item.create

Описание

`object item.create(object/array items)`

Этот метод позволяет создавать новые элементы данных.

Note:
Веб-элементы данных нельзя создавать через Zabbix API.

Note:
Этот метод доступен только пользователям типов *Admin* и *Super admin*. Разрешение на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Элементы данных для создания.

Помимо стандартных свойств элемента данных, метод принимает следующие параметры.

Parameter	Type	Description
preprocessing	array	Параметры предварительной обработки элемента данных.
tags	array	Теги элемента данных.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных элементов данных в свойстве `itemids`. Порядок возвращаемых идентификаторов соответствует порядку переданных элементов данных.

Примеры

Создание элемента данных

Создайте числовой элемент данных Zabbix agent с 2 тегами элемента данных, чтобы отслеживать свободное место на диске на узле сети с ID "30074".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
            {
                "tag": "equipment",
                "value": "workstation"
            }
        ],
        "delay": "30s"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Создание элемента данных инвентаря узла сети

Создайте элемент данных агента Zabbix, чтобы заполнить поле инвентаря узла сети "OS".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "uname",
        "key_": "system.uname",
        "hostid": "30021",
        "type": 0,
        "interfaceid": "30007",
        "value_type": 1,
        "delay": "10s",
        "inventory_link": 5
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Создание элемента данных с предварительной обработкой

Создайте элемент данных с использованием пользовательского множителя.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Создание зависимого элемента данных

Создайте зависимый элемент данных для мастер-элемента с ID 24759. Допускаются только зависимости на том же узле сети, поэтому мастер-элемент и зависимый элемент данных должны иметь одинаковый hostid.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
      "hostid": "30074",
      "name": "Dependent test item",
      "key_": "dependent.item",
      "type": 18,
```

```
        "master_itemid": "24759",
        "value_type": 2
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Создание элемента данных HTTP agent

Создайте элемент данных с методом запроса POST и предварительной обработкой JSON-ответа.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "url":"http://127.0.0.1/http.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value": "100"
            }
        ],
        "interfaceid": "1",
        "type": 19,
        "hostid": "10254",
        "delay": "5s",
        "key_": "json",
        "name": "HTTP agent example JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Создание элемента данных типа script

Создайте простой сбор данных с помощью элемента данных типа script.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.item",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Источник

CItem::create() в *ui/include/classes/api/services/CItem.php*.

item.delete

Описание

`object item.delete(array itemIds)`

Этот метод позволяет удалять элементы данных.

> **Note:**
> Веб-элементы данных нельзя удалить через Zabbix API.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых элементов данных.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных элементов данных, указанных в свойстве `itemids`.

Примеры

Удаление нескольких элементов данных

Удаление двух элементов данных.
 Зависимые элементы данных и прототипы элементов данных удаляются автоматически, если удаляется мастер-элемент данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Источник

CItem::delete() в *ui/include/classes/api/services/CItem.php*.

item.get

Описание

`integer/array item.get(object parameters)`

Метод позволяет получать элементы данных в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
itemids	ID/array	Возвращать только элементы данных с указанными ID.
groupids	ID/array	Возвращать только элементы данных, которые принадлежат узлам сети из указанных групп.
templateids	ID/array	Возвращать только элементы данных, которые принадлежат указанным шаблонам.
hostids	ID/array	Возвращать только элементы данных, которые принадлежат указанным узлам сети.
proxyids	ID/array	Возвращать только элементы данных, которые отслеживаются указанными прокси.
interfaceids	ID/array	Возвращать только элементы данных, которые используют указанные интерфейсы узлов сети.
graphids	ID/array	Возвращать только элементы данных, которые используются в указанных графиках.
triggerids	ID/array	Возвращать только элементы данных, которые используются в указанных триггерах.
webitems	flag	Включить web items в результат.
inherited	boolean	Если установлено в `true`, возвращать только элементы данных, унаследованные от шаблона.
inheritedTags	boolean	Возвращать элементы данных, у которых указанные `tags` также присутствуют в узле сети/шаблоне/связанных шаблонах. Possible values: `true` - шаблон/узел сети/связанные шаблоны также должны иметь указанные tags; `false` - *(default)* tags из шаблона/узла сети/связанных шаблонов игнорируются.
templated	boolean	Если установлено в `true`, возвращать только элементы данных, которые принадлежат шаблонам.
monitored	boolean	Если установлено в `true`, возвращать только включенные элементы данных, которые принадлежат отслеживаемым узлам сети.
group	string	Возвращать только элементы данных, которые принадлежат группе с указанным именем.
host	string	Возвращать только элементы данных, которые принадлежат узлу сети с указанным именем.
evaltype	integer	Метод оценки тегов. Possible values: 0 - *(default)* And/Or; 2 - Or.

Parameter	Type	Description
tags	array	Возвращать только элементы данных с указанными тегами. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все элементы данных. Possible operator values: 0 - *(default)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
with_triggers	boolean	Если установлено в `true`, возвращать только элементы данных, которые используются в триггерах.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, к которым принадлежит элемент данных.
selectInterfaces	query	Возвращать свойство `interfaces` с массивом интерфейсов узлов сети, используемых элементом данных.
selectTriggers	query	Возвращать свойство `triggers` с триггерами, в которых используется элемент данных. Supports `count`.
selectGraphs	query	Возвращать свойство `graphs` с графиками, которые содержат элемент данных. Supports `count`.

Parameter	Type	Description
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения элемента данных. Объект обнаружения элемента данных связывает обнаруженный элемент данных с прототипом элемента данных, из которого он был обнаружен. Он имеет следующие свойства: `parent_itemid` - (`string`) ID прототипа элемента данных, из которого был создан элемент данных; `key_` - (`string`) ключ прототипа элемента данных; `status` - (`int`) статус обнаружения элемента данных: 0 - *(default)* элемент данных обнаружен, 1 - элемент данных больше не обнаружен; `ts_delete` - (`timestamp`) время, когда элемент данных, который больше не обнаружен, будет удален; `ts_disable` - (`timestamp`) время, когда элемент данных, который больше не обнаружен, будет отключен; `disable_source` - (`int`) признак того, был ли элемент данных отключен правилом LLD или вручную: 0 - *(default)* отключен автоматически, 1 - отключен правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, которое создало элемент данных.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые присутствуют в шаблоне/узле сети/связанных шаблонах.
selectPreprocessing	query	Возвращать свойство `preprocessing` с параметрами предварительной обработки элемента данных.
selectTags	query	Возвращать теги элемента данных в свойстве `tags`.
selectValueMap	query	Возвращать свойство `valuemap` с картой значений элемента данных.

Parameter	Type	Description
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одно значение, либо массив значений для сравнения. Не поддерживает свойства с типом данных `text` data type. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, к которому принадлежит элемент данных.
limitSelects	integer	Ограничивает количество записей, возвращаемых подвыборками. Применяется к следующим подвыборкам: `selectGraphs` - результаты будут отсортированы по `name`; `selectTriggers` - результаты будут отсортированы по `description`.
sortfield	string/array	Сортировать результат по указанным свойствам. Possible values: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Эти параметры описаны в reference commentary.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	Возвращать свойство `itemDiscovery` с объектом обнаружения элемента данных. Объект обнаружения элемента данных связывает элемент данных с прототипом элемента данных, из которого он был создан. Этот запрос **устарел**, вместо него используйте `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;

- количество полученных объектов, если использован параметр `countOutput`.

Примеры

Поиск элементов данных по ключу

Получить все элементы данных, используемые в триггерах для указанного ID узла сети, у которых в ключе элемента данных есть слово "system.cpu", и отсортировать результаты по имени.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "CPU utilization",
            "key_": "system.cpu.util",
            "delay": "0",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "%",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42267",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "CPU utilization in %.",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "42264",
            "timeout": "",
            "url": "",
```

```
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "CPU utilization",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42259",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (15m avg)",
            "key_": "system.cpu.load[all,avg15]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42219",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
```

```
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "state": "0",
        "error": "",
        "name_resolved": "Load average (15m avg)",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "42249",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Load average (1m avg)",
        "key_": "system.cpu.load[all,avg1]",
        "delay": "1m",
        "history": "7d",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "42209",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "1",
        "description": "",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
```

```
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (1m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (5m avg)",
            "key_": "system.cpu.load[all,avg5]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
```

```
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (5m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "key_": "system.cpu.num",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
```

```
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Number of CPUs",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Поиск зависимых элементов данных по ключу

Получите все зависимые элементы данных с узла сети с ID ”10116”, у которых в ключе есть слово ”apache”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25550",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Days",
            "key_": "apache.status.uptime.days",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Days",
            "parameters": [],
```

```
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "25555",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Hours",
            "key_": "apache.status.uptime.hours",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
```

```
            "name_resolved": "Hours",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Поиск элемента данных HTTP agent

Найдите элемент данных HTTP agent с типом тела POST XML для указанного ID узла сети.

Запрос:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```


Ответ:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28252",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10255",
            "name": "template item",
            "key_": "ti",
            "delay": "30s",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
```

```
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "template item",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": ""
        }
    ],
    "id": 1
}
```

Получение элементов данных с правилами предварительной обработки

Получить все элементы данных и их правила предварительной обработки для указанного ID узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

См. также

- Правило обнаружения
- График
- Узел сети
- Интерфейс узла сети
- Триггер

Источник

CItem::get() в *ui/include/classes/api/services/CItem.php*.

item.update

Описание

`object item.update(object/array items)`

Этот метод позволяет обновлять существующие элементы данных.

Note:
Веб-элементы данных нельзя обновлять через Zabbix API.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Разрешение на вызов метода может быть отозвано в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства элемента данных, которые необходимо обновить.

Свойство `itemid` должно быть определено для каждого элемента данных, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Дополнительно к стандартным свойствам элемента данных, метод принимает следующие параметры.

Parameter	Type	Description
preprocessing	array	Параметры предобработки элемента данных для замены текущих параметров предобработки. Поведение параметра: - *read-only* для унаследованных объектов или обнаруженных объектов
tags	array	Теги элемента данных. Поведение параметра: - *read-only* для обнаруженных объектов

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных элементов данных, указанных в свойстве `itemids`.

Примеры

Включение элемента данных

Включите элемент данных, то есть установите для его статуса значение "0".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Обновление зависимого элемента данных

Обновите имя зависимого элемента данных и ID основного элемента данных. Допускаются только зависимости на одном и том же узле сети, поэтому основной и зависимый элемент данных должны иметь одинаковый hostid.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Обновление элемента данных HTTP-агента

Включить перехват значений элемента данных.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
```

```
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Обновление элемента данных с предварительной обработкой

Обновите элемент данных с правилом предварительной обработки элемента данных "In range".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Обновление элемента данных типа скрипт

Обновите элемент данных типа скрипт, указав другой скрипт, и удалите ненужные параметры, которые использовались предыдущим скриптом.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
```

```
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Источник

CItem::update() в *ui/include/classes/api/services/CItem.php*.

Элемент данных графика

Этот класс предназначен для работы с элементами данных графика.

Ссылки на объекты:

- Элемент данных графика

Доступные методы:

- graphitem.get - получить элементы данных графика

Объект элемента данных графика

Следующие объекты напрямую связаны с API `graphitem`.

Элемент графика

Note:
Элементы графика можно изменять только через API `graph`.

Объект элемента графика имеет следующие свойства.

Property	Type	Description
gitemid	ID	ID элемента графика. Поведение свойства: - *только для чтения*
color	string	Цвет отрисовки элемента графика в виде шестнадцатеричного кода цвета. Поведение свойства: - *обязательно* для операций создания
itemid	ID	ID элемента данных. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
calc_fnc	integer	Значение элемента данных, которое будет отображаться. Возможные значения: 1 - минимальное значение; 2 - *(по умолчанию)* среднее значение; 4 - максимальное значение; 7 - все значения; 9 - последнее значение, используется только для круговых и развернутых графиков.
drawtype	integer	Стиль отображения элемента графика. Возможные значения: 0 - *(по умолчанию)* линия; 1 - закрашенная область; 2 - жирная линия; 3 - точка; 4 - пунктирная линия; 5 - градиентная линия.
graphid	ID	ID графика, которому принадлежит элемент графика.
sortorder	integer	Позиция элемента на графике. По умолчанию: начинается с ”0” и увеличивается на единицу с каждой записью.
type	integer	Тип элемента графика. Возможные значения: 0 - *(по умолчанию)* простой; 2 - сумма графика, используется только для круговых и развернутых графиков.
yaxisside	integer	Сторона графика, на которой будет отображаться шкала Y элемента графика. Возможные значения: 0 - *(по умолчанию)* левая сторона; 1 - правая сторона.

graphitem.get

Описание

`integer/array graphitem.get(object parameters)`

Метод позволяет получить элементы графика в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие требуемый результат.

Метод поддерживает следующие параметры.

Parameter	Type	Description
graphids	ID/array	Возвращать только элементы графиков, которые принадлежат указанным графикам.
itemids	ID/array	Возвращать только элементы графиков с указанными ID элементов данных.

Parameter	Type	Description
type	integer	Возвращать только элементы графиков с указанным типом. См. список поддерживаемых типов элементов графиков на странице объекта элемента графика.
selectGraphs	query	Возвращать свойство `graphs` с массивом графиков, к которым принадлежит элемент данных.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `gitemid`.
countOutput	boolean	Эти параметры описаны в справочном комментарии.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение элементов графика из графика

Получить все элементы графика, используемые в графике, с дополнительной информацией об элементе данных и узле сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "gitemid": "1242",
            "graphid": "387",
            "itemid": "22665",
            "drawtype": "1",
            "sortorder": "1",
            "color": "FF5555",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1243",
            "graphid": "387",
            "itemid": "22668",
            "drawtype": "1",
            "sortorder": "2",
            "color": "55FF55",
```

```
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1244",
            "graphid": "387",
            "itemid": "22671",
            "drawtype": "1",
            "sortorder": "3",
            "color": "009999",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        }
    ],
    "id": 1
}
```


Смотрите также

- График

Источник

CGraphItem::get() в *ui/include/classes/api/services/CGraphItem.php*.

Приложение 1. Справочные комментарии

Обозначение Сигнатура метода

Каждая страница метода API содержит сигнатуру метода.

Пример: `object action.create(object/array actions)`

Сигнатура	Описание
`object`	Тип данных, возвращаемых методом.
`action.create`	Полное имя метода.
`object/array`	Тип данных, принимаемых методом.
`actions`	Краткое описание данных, принимаемых методом.

Типы данных

Zabbix API поддерживает следующие типы данных в качестве входных данных:

Type	Description
ID	Уникальный идентификатор, используемый для ссылки на сущность.
boolean	Булево значение (либо `true`, либо `false`).
flag	Значение, которое считается `true`, если оно передано и не равно `null`; в противном случае значение считается `false`.
integer	Целое число.
float	Число с плавающей точкой.
string	Текстовая строка.
text	Более длинная текстовая строка.
timestamp	Метка времени Unix.
array	Упорядоченная последовательность значений (обычный массив).
object	Ассоциативный массив.

Type	Description
query	Значение, определяющее, какие данные должны быть возвращены. Значение может быть задано как массив имен свойств (чтобы вернуть только определенные свойства) или как одно из предопределенных значений: `extend` - возвращает все свойства объекта; `count` - возвращает количество полученных записей, поддерживается только некоторыми вложенными выборками.

Attention:
Zabbix API всегда возвращает значения только в виде строк или массивов.

Поведение свойства

Некоторые свойства объектов помечены короткими метками, описывающими их поведение. Используются следующие обозначения:

- ***только для чтения*** - значение свойства задаётся автоматически и не может быть определено или изменено пользователем, даже в особых случаях (например, *только для чтения* для наследуемых или обнаруженных объектов);
- ***только для записи*** - значение свойства можно установить, но нельзя прочитать после;
- ***постоянное*** -значение свойства можно задать при создании объекта, но нельзя изменить позже;
- ***поддерживаемое*** - значение свойства не обязательно к установке, но может быть задано при определённых условиях (например, *поддерживаемое*, если `type` установлен в ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”). Однако даже *поддерживаемое* свойства могут автоматически принимать значения по умолчанию независимо от условий;
- ***обязательное*** значение свойства обязательно для указания при всех операциях (кроме get) или при определённых условиях (например, *обязательное* при создании объекта; *обязательное*, если `operationtype` имеет значение ”global script”, а `opcommand_hst` не задан).

Note:
Для операций обновления свойство считается ”установленным”, если его значение задаётся в процессе выполнения операции обновления.

Свойства, не отмеченные метками, являются необязательными.

Поведение параметров

Некоторые параметры операций помечены короткими метками, описывающими их поведение для операции. Используются следующие метки:

- ***read-only*** - значение параметра задается автоматически и не может быть определено или изменено пользователем, даже при некоторых особых условиях (например, *read-only* для унаследованных объектов или обнаруженных объектов);
- ***write-only*** - значение параметра можно задать, но после этого к нему нельзя получить доступ;
- ***supported*** - значение параметра не требуется задавать, но его можно задать при некоторых особых условиях (например, *supported*, если `operating_mode` объекта Proxy установлен в ”passive proxy”); однако следует отметить, что параметры *supported* по-прежнему могут быть установлены в значения по умолчанию независимо от условий;
- ***required*** - значение параметра обязательно должно быть задано.

Параметры, не помеченные метками, являются необязательными.

Зарезервированное значение ID равное ”0” Зарезервированное значение ID ”0” можно использовать для фильтрации элементов и удаления связанных объектов. Например, для удаления ссылки на прокси с узла сети, proxytid необходимо задать значением 0 (”proxyid”: ”0”) или для фильтрации узлов сети наблюдаемых сервером, опция proxyids должна быть задана значением 0 (”proxyids”: ”0”).

Общие параметры метода `get` Следующие параметры поддерживаются всеми методами `get`:

Parameter	Type	Description
countOutput	boolean	Возвращать количество записей в результате вместо фактических данных.

Parameter	Type	Description
editable	boolean	Если установлено значение `true`, возвращать только те объекты, на которые у пользователя есть права на запись. По умолчанию: `false`.
excludeSearch	boolean	Возвращать результаты, которые не соответствуют критериям, заданным в параметре `search`.
filter	object	Возвращать только те результаты, которые в точности соответствуют заданному фильтру. Принимает объект, где ключи — это имена свойств (например, свойства объекта Host в `host.get`, свойства объекта Item в `item.get` и т. д.), а значения — либо одно значение, либо массив значений, с которыми выполняется сравнение. Не поддерживает свойства с типом данных `text`. Обратите внимание, что некоторые методы имеют специальную функциональность для этого параметра, которая описана на странице метода (например, параметр `filter` в host.get также поддерживает свойства интерфейса узла сети).
limit	integer	Ограничить количество возвращаемых записей.
output	query	Свойства объекта, которые нужно вернуть. Обратите внимание, что идентификатор объекта (то есть `hostid`, `itemid` и т. д.) всегда включается в ответ, даже если он не указан в параметре `output`. По умолчанию: `extend`.
preservekeys	boolean	Использовать идентификаторы в качестве ключей в результирующем массиве.

Parameter	Type	Description
search	object	Возвращать результаты, которые соответствуют заданному шаблону (без учета регистра). Принимает объект, где ключи — это имена свойств (например, свойства объекта Host в `host.get`, свойства объекта Item в `item.get` и т. д.), а значения — строки для поиска. Если дополнительные параметры не заданы, будет выполнен поиск `LIKE "%…%"`. Поддерживает только свойства с типом данных `string` и `text`. Обратите внимание, что некоторые методы имеют специальную функциональность для этого параметра, которая описана на странице метода (например, параметр `search` в host.get также поддерживает свойства интерфейса узла сети).
searchByAny	boolean	Если установлено значение `true`, возвращать результаты, которые соответствуют любому из критериев, заданных в параметре `filter` или `search`, а не всем сразу. По умолчанию: `false`.
searchWildcardsEnabled	boolean	Если установлено значение `true`, позволяет использовать `*` в качестве символа подстановки в параметре `search`. По умолчанию: `false`.
sortfield	string/array	Сортировать результат по заданным свойствам. См. описание конкретного метода API `get` для списка свойств, которые можно использовать для сортировки. Макросы не раскрываются перед сортировкой. Если значение не указано, данные будут возвращены без сортировки.
sortorder	string/array	Порядок сортировки. Если передан массив, каждое значение будет сопоставлено соответствующему свойству, указанному в параметре `sortfield`. Возможные значения: `ASC` - *(по умолчанию)* по возрастанию; `DESC` - по убыванию.

Parameter	Type	Description
startSearch	boolean	Параметр `search` будет сравнивать начало полей, то есть вместо этого выполнять поиск `LIKE "…%"`. Игнорируется, если `searchWildcardsEnabled` установлено в `true`.

Флаги происхождения сущности Методы получения возвращают свойство `flags` для сущностей, связанных с низкоуровневым обнаружением (LLD-правило/прототип LLD-правила, элемент данных/прототип элемента данных и т. д.). Это свойство полезно для обозначения того, была ли сущность обнаружена или нет, поскольку возможности редактирования для обнаруженных сущностей ограничены.

Свойство `flags` возвращает результат на основе комбинации (операция ”+”) следующих значений:

Value	Description
0	Базовая сущность (элемент данных, триггер, график, узел сети)
1	Правило низкоуровневого обнаружения
2	Любой прототип (прототип элемента данных, прототип триггера, прототип LLD-правила и т. д.)
4	Обнаруженная сущность (обнаруженный элемент данных, триггер, график, узел сети, LLD-правило)

Комбинированное значение, возвращаемое свойством `flags`, может быть следующим:

Value	Combination of	Description
0	0	Обычная сущность (элемент данных, триггер, график, узел сети).
2	2	Прототип сущности (прототип элемента данных, прототип триггера и т. д.).
6	2+4	Обнаруженный элемент данных, триггер, график, узел сети (преобразован из прототипа).
1	1	Правило низкоуровневого обнаружения.
3	1+2	Прототип правила низкоуровневого обнаружения.
5	1+4	Обнаруженное правило низкоуровневого обнаружения (преобразовано из прототипа).
7	1+2+4	Обнаруженный прототип правила низкоуровневого обнаружения.

Примеры Проверка прав пользователя

Имеет ли пользователь разрешение на запись на хосты, имена которых начинаются с ”MySQL” или ”Linux”?

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
Нулевой результат означает отсутствие хостов с разрешениями на чтение/запись.

Подсчет несоответствий

Подсчитать количество хостов, имена которых не содержат подстроку ”ubuntu”

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Поиск хостов по подстановочным знакам

Найти хосты, имя которых содержит слово «server» и имеют интерфейсные порты «10050» или «10071». Отсортировать результат по имени хоста в порядке убывания и ограничьте его до 5 хостов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    ],
    "id": 1
}
```


Поиск хостов с использованием подстановочных знаков с «preservekeys»

Если к предыдущему запросу добавить параметр «preservekeys», то результат возвращается в виде ассоциативного массива, где ключами являются id объектов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
```

```
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5,
        "preservekeys": true
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "50003": {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50005": {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50004": {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "99032": {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        "99061": {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
```

```
                    "port": "10050"
                }
            ]
        }
    },
    "id": 1
}
```

Приложение 2. Изменения с 7.4 до 8.0

Изменения, несовместимые с предыдущими версиями

Другие изменения и исправления ошибок

Приложение 3. Изменения в 8.0

8.0.1 Эта версия ещё не выпущена.

Расширения

Обзор Хотя Zabbix предлагает множество возможностей, всегда есть место для дополнительных функций. Расширения — это удобный способ изменения и увеличения возможностей мониторинга при помощи Zabbix без изменения исходного кода.

Вы можете расширить функциональность Zabbix как используя встроенные возможности расширения (траппер элементы данных, пользовательские параметры и т. д.), так и путём использования или создания пользовательских расширений (загружаемых модулей, плагинов и т. д.).

В этом разделе представлен обзор со ссылками на все варианты расширения Zabbix.

Сбор данных с помощью пользовательских команд Элементы данных типа Trapper

Элементы данных типа Trapper — это элементы данных, которые принимают входящие данные вместо того, чтобы запрашивать их.

Элементы данных типа Trapper полезны для отправки определенных данных на сервер Zabbix или прокси, например периодических данных о доступности и производительности в случае длительно выполняемых пользовательских скриптов.

Отправка данных на сервер Zabbix или прокси возможна с помощью утилиты Zabbix sender или протокола Zabbix sender.

Отправка данных на сервер Zabbix также возможна с помощью метода API `history.push`.

Внешние проверки

Внешняя проверка — это элемент данных для выполнения проверок путём запуска исполняемого файла, например сценария оболочки или двоичного файла.

Внешние проверки выполняются Zabbix сервером или прокси (когда узел сети наблюдается через прокси) и не требуют наличия агента, работающего на отслеживаемом узле сети.

Пользовательские параметры

Пользовательский параметр — это заданная пользователем команда (связанная с задаваемым пользователем ключом), которая при выполнении может получить необходимые вам данные с узла сети, на котором работает Zabbix агент. Пользовательские параметры полезны для настройки элементов данных агента или агента 2, отсутствующих в Zabbix.

Элементы данных Zabbix agent `system.run[]`

`system.run[]` — это элемент данных агента Zabbix для пользовательской команды (связанной с предопределённым ключом `system.run[]`, например, `system.run[myscript.sh]`), которую можно выполнить на узле сети, где запущен агент Zabbix.

> **Note:**
> Элементы данных `system.run[]` по умолчанию отключены. При использовании их необходимо разрешить (разрешённые) и определить в файле конфигурации агента Zabbix или Zabbix agent 2 с помощью параметра конфигурации (`AllowKey` или `AllowKeyRegexp`).

> **Attention:**
> Пользовательские команды в элементах данных, таких как внешние проверки, пользовательские параметры и элементы данных Zabbix агента `system.run[]`, выполняются от того пользователя ОС, который используется для запуска компонентов Zabbix. Для выполнения этих команд данный пользователь должен обладать необходимыми правами.

Элементы данных типа HTTP agent

Элемент данных типа HTTP agent предназначен для выполнения запросов данных по HTTP/HTTPS. Элементы данных типа HTTP agent полезны для отправки запросов к конечным точкам HTTP с целью получения данных из таких сервисов, как *Elasticsearch* и *OpenWeatherMap*, проверки состояния API Zabbix, веб-сервера Apache или NGINX и т. д. Элементы данных типа HTTP agent (с включенной обработкой входящих данных) также могут функционировать как элементы данных типа trapper.

Скриптовые элементы данных

Скриптовый элемент данных — это элемент данных для выполнения пользовательского кода JavaScript, который получает данные через HTTP/HTTPS. Скриптовые элементы данных полезны, когда функциональности, предоставляемой элементами HTTP агента, недостаточно. Например, в сложных сценариях сбора данных, требующих нескольких шагов или сложной логики, скриптовый элемент данных можно настроить для выполнения HTTP-вызова, а затем обработки полученных данных и последующей передачи преобразованного значения второму HTTP-вызову.

> **Note:**
> Элементы данных HTTP агента и скриптовые элементы данных поддерживаются Zabbix сервером и прокси сервером, и не требуют наличия агента, работающего на отслеживаемом узле сети.

Продвинутые расширения Подгружаемые модули

Подгружаемые модули, написанные на языке C, представляют собой универсальный и ориентированный на производительность вариант расширения функциональности компонентов Zabbix (сервер, прокси, агент) на платформах UNIX. Подгружаемый модуль — это, по сути, общая библиотека, используемая демоном Zabbix и подгружаемая при запуске. Библиотека должна содержать определённые функции, чтобы процесс Zabbix мог обнаружить, что файл действительно является модулем, который он может загрузить и с которым он может работать.

Подгружаемые модули имеют набор преимуществ, включая возможность добавлять новые метрики или реализовывать любую другую логику (для примера, экспорта данных истории Zabbix), прекрасную производительность и возможность разрабатывать, использовать и делиться функциональными возможностями, которые они предоставляют. Это способствует бесперебойному обслуживанию и помогает предоставлять новые функциональные возможности проще и независимо от основной кодой базы Zabbix.

Подгружаемые модули особенно полезны при сложной настройке мониторинга. При мониторинге встроенных систем наличие большого количества отслеживаемых параметров или тяжелые сценарии со сложной логикой или длительным временем запуска такие расширения, как пользовательские параметры, элементы Zabbix агента `system.run[]` и внешние проверки, будут влиять на производительность. Загружаемые модули предлагают способ расширения функциональности Zabbix без ущерба для производительности.

Плагины

Плагины расширяют возможности мониторинга Zabbix agent 2. Написанные на языке программирования Go, они представляют собой альтернативу загружаемым модулям (написанным на C).

Можно разработать собственный плагин (независимый сборщик данных) для Zabbix agent 2. См. Developer center.

Настройка оповещений Вебхуки

Вебхук — это способ оповещения Zabbix, который позволяет расширить возможности Zabbix отсылать оповещения во внешнее программное обеспечение, такое как системы службы поддержки, чаты или мессенджеры. Подобно скриптовым

элементам данных, вебхуки полезны для выполнения HTTP-вызовов с использованием пользовательского кода JavaScript, например, для отправки уведомлений на различные платформы, такие как Microsoft Teams, Discord и Jira. Также возможно возвращать некоторые данные (например, о созданных заявках в службу поддержки), которые затем отображаются в Zabbix.

Существующие вебхуки доступны в Git репозитории Zabbix. Для разработки собственного вебхука обратитесь к Руководству по разработке вебхуков [en].

Скрипты оповещений

Скрипт оповещения — это способ оповещения Zabbix, который предоставляет возможность создать альтернативный способ (скрипт) для управления оповещениями Zabbix. Скрипты оповещений полезны, если вас не устраивают существующие способы оповещения для отсылки уведомлений в Zabbix.

Кастомизация веб-интерфейса Пользовательские темы

Можно изменить внешний вид веб-интерфейса Zabbix, используя пользовательские темы. Смотрите инструкции по созданию и применению своих собственных тем.

Модули веб-интерфейса

Модули веб-интерфейса предоставляют возможность расширить функциональность веб-интерфейса Zabbix путем добавления сторонних модулей или разработки своих собственных. С помощью модулей веб-интерфейса вы можете добавлять новые пункты меню, соответствующие им представления, действия и т.д.

Глобальные скрипты Глобальный скрипт — это определяемый пользователем набор команд, которые могут быть выполнены на объекте мониторинга (с помощью интерпретатора оболочки (/bin/sh)), в зависимости от настроенной области и прав пользователя. Глобальные скрипты можно настроить для следующих действий:

- Операция действия
- Действие вручную над узлом сети
- Действие вручную над событиями

Глобальные скрипты полезны во многих случаях. Например, если настроены операции действий или действия вручную над узлом сети, вы можете использовать глобальные скрипты для автоматического или ручного выполнения удалённых команд, например, перезапуск приложения (веб-сервера, промежуточного программного обеспечения, CRM и т. д.) или освобождение места на диске (удаление старых файлов, очистка `/tmp` и т. д.). Или другой пример: если настроено действие вручную над событиями, вы можете использовать глобальные скрипты для управления заявками о проблемах во внешних системах.

Глобальные скрипты могут выполняться Zabbix сервером, прокси или агентом.

> **Attention:**
> Пользовательские команды выполняются от пользователя ОС, который используется для запуска компонентов Zabbix. Для выполнения этих команд данный пользователь должен обладать необходимыми правами.

Zabbix API Zabbix API — это API на основе HTTP, который является частью веб-интерфейса Zabbix. С помощью Zabbix API вы можете выполнять любые из следующих операций:

- Программное получение и изменение конфигурации Zabbix.
- Импорт и экспорт конфигурации Zabbix.
- Доступ к данным истории и динамики изменений Zabbix.
- Настройка приложений для работы с Zabbix.
- Интеграция Zabbix со сторонним программным обеспечением.
- Автоматизация рутинных задач.

Zabbix API состоит из множества методов, которые номинально сгруппированы в отдельные API. Каждый метод выполняет определённую задачу. Доступные методы, а также обзор функций, предоставляемых Zabbix API, смотрите в Справочнике методов Zabbix API.

Загружаемые модули

Обзор

Подгружаемые модули предлагают ориентированный на производительность вариант расширения Zabbix.

Функциональность Zabbix можно расширить многими способами, например, с помощью пользовательских параметров, внешних проверок, а также элементов данных Zabbix агента system.run. Они работают очень хорошо, но имеют один большой недостаток, называемый fork(). Zabbix для сбора пользовательских проверок каждый раз должен создавать новый порождённый процесс, что не очень хорошо сказывается на производительности. Обычно это не самая большая проблема, но тем не менее это может быть серьезной проблемой в случае мониторинга встроенных систем, имеющих большое количество наблюдаемых параметров или тяжёлых скриптов со сложной логикой или длительным временем запуска.

Поддержка подгружаемых модулей предлагает пути расширения Zabbix агента, сервера и прокси без ущерба для производительности.

Подгружаемый модуль — в своей основе разделяемая библиотека, которая используется Zabbix демоном и подгружается при старте демона. Библиотека должна содержать определённые функции, чтобы Zabbix процесс мог определить, что файл на самом деле является модулем, который можно загрузить и с ним работать.

Подгружаемые модули имеют много преимуществ. Отличная производительность и возможность реализовать любую логику очень важны, но, возможно, наиболее важное преимущество — возможность разработать, использовать и поделиться Zabbix модулем. Это способствует беспроблемному обслуживанию и помогает вносить новые функции легче и независимо от основной кодовой базы Zabbix.

Лицензирование и распространение модулей в бинарной форме регламентируется лицензией AGPL-3.0 (модули линкуются с Zabbix во время выполнения и используют заголовки Zabbix; в настоящее время весь код Zabbix, начиная с версии 7.0, лицензируется под лицензией AGPL-3.0). Бинарная совместимость не гарантируется Zabbix'ом.

Постоянство API модулей гарантируется в пределах одного цикла выпуска Zabbix LTS (Долгосрочная поддержка). Постоянство Zabbix API не гарантируется (технически имеется возможность вызова внутренних функций Zabbix из модуля, но нет гарантии, что такие модули будут работать).

API модулей

Для того чтобы разделяемая библиотека обрабатывалась как Zabbix модуль, она должна реализовывать и экспортировать несколько функций. На данный момент имеется шесть функций в API модулей Zabbix, только одна из которых обязательна, а остальные пять — не обязательны.

Обязательный интерфейс

Единственной обязательной функцией является `zbx_module_api_version()`:

```
int zbx_module_api_version(void);
```

Эта функция должна возвращать версию API, реализованную этим модулем, и для того чтобы модуль был загружен, эта версия должна совпадать с версией API модуля, поддерживаемой Zabbix. Версия API модуля, поддерживаемая Zabbix, — `ZBX_MODULE_API_VERSION`. Поэтому эта функция должна возвращать именно эту константу. Старая константа `ZBX_MODULE_API_VERSION_ONE`, ранее использовавшаяся для этой цели, теперь определена как равная `ZBX_MODULE_API_VERSION` для сохранения совместимости с исходным кодом, однако ее использование не рекомендуется.

Необязательный интерфейс

Необязательными функциями являются `zbx_module_init()`, `zbx_module_item_list()`, `zbx_module_item_timeout()`, `zbx_module_history_write_cbs()` и `zbx_module_uninit()`:

```
int zbx_module_init(void);
```

Эта функция должна выполнить необходимую инициализацию модуля, если она требуется. При успешном выполнении она должна вернуть `ZBX_MODULE_OK`. В противном случае она должна вернуть `ZBX_MODULE_FAIL`. В последнем случае Zabbix не запустится.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Эта функция должна возвращать список элементов данных, поддерживаемых модулем. Каждый элемент данных определяется в структуре `ZBX_METRIC`, см. раздел ниже для подробностей. Список завершается структурой `ZBX_METRIC` с полем "key", равным NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Если модуль экспортирует `zbx_module_item_list()`, то эта функция используется Zabbix для указания настроек тайм-аута в файле конфигурации Zabbix, которым должны руководствоваться проверки элементов данных, реализованные модулем. Здесь параметр "timeout" задается в секундах.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Эта функция должна возвращать функции обратного вызова, которые сервер Zabbix будет использовать для экспорта истории различных типов данных. Функции обратного вызова предоставляются в виде полей структуры ZBX_HISTORY_WRITE_CBS; поля могут быть NULL, если модуль не заинтересован в истории определенного типа.

```
int zbx_module_uninit(void);
```

Эта функция должна выполнить необходимую деинициализацию, если она требуется, например освобождение выделенных ресурсов, закрытие файловых дескрипторов и т. д.

Все функции вызываются один раз при запуске Zabbix, когда модуль загружается, за исключением zbx_module_uninit(), которая вызывается один раз при завершении работы Zabbix, когда модуль выгружается.

Определение элементов данных

Каждый элемент данных определяется в структуре ZBX_METRIC:

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Здесь **key** — это ключ элемента данных (например, "dummy.random"), **flags** — либо CF_HAVEPARAMS, либо 0 (в зависимости от того, принимает элемент данные параметры или нет), **function** — это функция C, реализующая элемент данных (например, zbx_module_dummy_random), а **test_param** — это список параметров, который будет использоваться при запуске агента Zabbix с флагом -p (например, "1,1000", может быть NULL). Пример определения может выглядеть так:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Каждая функция, реализующая элемент данных, должна принимать два параметра-указателя: первый типа AGENT_REQUEST, а второй типа AGENT_RESULT:

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

Эти функции должны возвращать SYSINFO_RET_OK, если значение элемента данных было успешно получено. В противном случае они должны возвращать SYSINFO_RET_FAIL. См. пример модуля "dummy" ниже, чтобы узнать, как получать информацию из AGENT_REQUEST и как задавать информацию в AGENT_RESULT.

Предоставление callback-функций для экспорта истории

> **Attention:**
> Экспорт истории через модуль больше не поддерживается прокси Zabbix.

Модуль может задавать функции для экспорта данных истории по типам: Numeric (float), Numeric (unsigned), Character, Text и Log:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
```

```
}
ZBX_HISTORY_WRITE_CBS;
```

Каждая из них должна принимать в качестве аргументов массив `history` из `history_num` элементов. В зависимости от типа данных истории, которые требуется экспортировать, `history` представляет собой массив следующих структур соответственно:

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    double      value;
}
ZBX_HISTORY_FLOAT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
    const char  *source;
    int     timestamp;
    int     logeventid;
    int     severity;
}
ZBX_HISTORY_LOG;
```

Callback-функции будут использоваться процессами синхронизации истории сервера Zabbix в конце процедуры синхронизации истории после записи данных в базу данных Zabbix и сохранения в кэше значений.

Attention:
В случае внутренней ошибки в модуле экспорта истории рекомендуется реализовать модуль таким образом, чтобы он не блокировал весь мониторинг до восстановления, а вместо этого отбрасывал данные и позволял серверу Zabbix продолжать работу.

Сборка модулей

В настоящее время модули предназначены для сборки внутри дерева исходного кода Zabbix, поскольку API модуля зависит от некоторых структур данных, определенных в заголовочных файлах Zabbix.

Наиболее важный заголовочный файл для загружаемых модулей — `include/module.h`, который определяет эти структуры данных. Другие необходимые системные заголовочные файлы, которые помогают `include/module.h` работать корректно, — это `stdlib.h` и `stdint.h`.

С учетом этой информации все готово для сборки модуля. Модуль должен включать `stdlib.h`, `stdint.h` и `module.h`, а сценарий сборки должен гарантировать, что эти файлы находятся в пути включения. Подробности см. в примере модуля "dummy" ниже.

Еще один полезный заголовочный файл — `include/zbxcommon.h`, который определяет функцию `zabbix_log()`, которую можно использовать для ведения журнала и отладки.

Параметры конфигурации

Агент Zabbix, сервер и прокси поддерживают два параметра для работы с модулями:

- `LoadModulePath` - полный путь к расположению загружаемых модулей
- `LoadModule` - модуль или модули, загружаемые при запуске.

Модули должны находиться в каталоге, указанном в `LoadModulePath`, либо перед именем модуля должен быть указан путь. Если указанный путь является абсолютным (начинается с `/`), параметр `LoadModulePath` игнорируется. Можно указать несколько параметров `LoadModule`.

Например, чтобы расширить возможности агента Zabbix, можно добавить следующие параметры:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

При запуске агент загрузит модули mariadb.so, apache.so и kernel.so из каталога `/usr/local/lib/zabbix/agent`, а модуль dummy.so будет загружен из `/usr/local/lib/zabbix`. Агент не запустится, если модуль отсутствует, заданы некорректные разрешения или общая библиотека не является модулем Zabbix.

Конфигурация веб-интерфейса

Загружаемые модули поддерживаются агентом Zabbix, сервером и прокси. Поэтому тип элемента данных в веб-интерфейсе Zabbix зависит от того, куда загружен модуль. Если модуль загружен в агент, то тип элемента данных должен быть "Zabbix agent" или "Zabbix agent (active)". Если модуль загружен в сервер или прокси, то тип элемента данных должен быть "Simple check".

Экспорт истории через модули Zabbix не требует какой-либо конфигурации веб-интерфейса. Если модуль успешно загружен сервером и предоставляет функцию `zbx_module_history_write_cbs()`, которая возвращает как минимум одну ненулевую callback-функцию, то экспорт истории будет включен автоматически.

Модуль dummy

Zabbix включает пример модуля, написанного на языке C. Модуль расположен в `src/modules/dummy`:

```
alex@alex:~trunk/src/modules/dummy$ ls -l
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c
-rw-rw-r-- 1 alex alex   67 Apr 24 17:54 Makefile
-rw-rw-r-- 1 alex alex  245 Apr 24 17:54 README
```

Модуль хорошо документирован, его можно использовать как шаблон для собственных модулей.

После выполнения `./configure` в корне дерева исходного кода Zabbix, как описано выше, просто выполните `make`, чтобы собрать `dummy.so`.

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
```

```
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/*                    -                     */
static int  item_timeout = 0;

/*                              static                    ,              */
/*      Zabbix (zbx_*)       API            (zbx_module_*)                          */
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
```

```
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
```

```
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
```

```
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
```

```
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

Модуль экспортирует три новых элемента данных:

- `dummy.ping` - всегда возвращает '1'
- `dummy.echo[param1]` - возвращает первый параметр как есть, например `dummy.echo[ABC]` вернет ABC
- `dummy.random[param1, param2]` - возвращает случайное число в диапазоне от param1 до param2, например `dummy.random[1,1000000]`

Ограничения

Поддержка загружаемых модулей реализована только для платформы Unix. Это означает, что она не работает для агентов Windows.

В некоторых случаях модулю может потребоваться считывать параметры конфигурации, связанные с модулем, из `zabbix_agentd.conf`. В настоящее время это не поддерживается. Если вашему модулю нужно использовать какие-либо параметры конфигурации, вам, вероятно, следует реализовать разбор отдельного файла конфигурации, специфичного для модуля.

Модули веб-интерфейса

Обзор

Функциональность веб-интерфейса Zabbix можно расширить, добавляя сторонние модули или разрабатывая собственные модули без необходимости изменять исходный код Zabbix.

Обратите внимание, что код модуля будет выполняться с теми же привилегиями, что и исходный код Zabbix. Это означает следующее:

- Сторонние модули могут быть вредоносными. Вы должны доверять модулям, которые устанавливаете.
- Ошибки в коде стороннего модуля могут привести к сбою веб-интерфейса. Если это произойдет, просто удалите код модуля из веб-интерфейса. Как только вы перезагрузите веб-интерфейс Zabbix, вы увидите сообщение о том, что некоторые модули отсутствуют. Перейдите в Управление модулями (в *Administration* > *General* > *Modules*) и снова нажмите *Scan directory*, чтобы удалить несуществующие модули из базы данных.

Установка

Всегда читайте руководство по установке для конкретного модуля. Рекомендуется устанавливать новые модули по одному, чтобы легче выявлять сбои.

Непосредственно перед установкой модуля:

- Убедитесь, что вы загрузили модуль из надежного источника. Установка вредоносного кода может привести к последствиям, таким как потеря данных.
- Разные версии одного и того же модуля (с одним и тем же ID) можно устанавливать параллельно, но одновременно может быть включена только одна версия.

Шаги по установке модуля:

- Распакуйте модуль в отдельную папку в каталоге `modules` установки веб-интерфейса Zabbix (например, `zabbix/ui/modules`).
- Убедитесь, что папка модуля содержит как минимум файл `manifest.json`.
- Перейдите в Администрирование модулей и нажмите кнопку *Сканировать каталог*.
- Новый модуль появится в списке вместе со своей версией, автором, описанием и статусом.
- Включите модуль, нажав на его статус.

Устранение неполадок:

Проблема	Решение
Модуль не появился в списке	Если ваш модуль не появился в списке, убедитесь, что `manifest.json` существует в `modules/your-module/`. Если его нет, возможно, вы распаковали модуль не в тот каталог. Если файл есть, модуль может быть несовместим с вашей версией Zabbix. Также проверьте, что пользователь веб-сервера имеет как минимум права на чтение и поиск (`r-x`) для папки модуля и всех подкаталогов, а также права на чтение (`r--`) для всех файлов внутри.
Веб-интерфейс аварийно завершил работу	Код модуля несовместим с текущей версией Zabbix или конфигурацией сервера. Удалите файлы модуля и перезагрузите веб-интерфейс. Вы увидите уведомление о том, что некоторые модули отсутствуют. Перейдите в Администрирование модулей и снова нажмите *Сканировать каталог*, чтобы удалить несуществующие модули из базы данных.
Появляется сообщение об ошибке с одинаковым пространством имен, ID или действиями	Новый модуль попытался зарегистрировать пространство имен, ID или действия, которые уже зарегистрированы другими включенными модулями. Отключите конфликтующий модуль, указанный в сообщении об ошибке, перед включением нового.
Появляются технические сообщения об ошибках	Сообщите об ошибках разработчику модуля.

Разработка модулей

Информацию о разработке пользовательских модулей смотрите в Центре разработчиков [en].

Приложения

Обзор В этом разделе содержатся справочные материалы, параметры конфигурации, технические спецификации и руководства по процедурам, дополняющие основное руководство.

Основной справочник

- Установка и настройка — создание базы данных, установка и настройка сервера и агента.
- Конфигурация демонов — параметры конфигурации для Zabbix сервера, прокси, агента и так далее.
- Элементы данных — справочник по ключам элементов данных и поддерживаемым типам данных.
- Макросы — синтаксис и применение встроенных и пользовательских макросов.

Установка и настройка

В этом разделе рассматриваются создание базы данных, настройка безопасности, процедуры настройки для конкретных платформ и конфигурация функций.

Пожалуйста, используйте боковую панель, чтобы найти процедуру, соответствующую вашей среде.

Создание базы данных

Обзор

База данных Zabbix должна быть создана во время установки сервера Zabbix или прокси.

В этом разделе приведены инструкции по созданию базы данных Zabbix. Для каждой поддерживаемой базы данных доступен отдельный набор инструкций.

> **Note:**
> Чтобы повысить безопасность базы данных путем создания ролей/пользователей базы данных с минимальными привилегиями, см. рекомендации по созданию базы данных для каждой поддерживаемой базы данных:

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
>
> Для настройки защищенных TLS-соединений см. Защищенное соединение с базой данных.

UTF-8 — единственная кодировка, поддерживаемая Zabbix. Известно, что она работает без каких-либо проблем с безопасностью. Пользователям следует учитывать, что при использовании некоторых других кодировок известны проблемы безопасности. Для перехода на UTF-8 см. Исправление набора символов и сопоставления базы данных Zabbix. Также см. Ограничения фильтрации с сопоставлениями utf8mb4.

> **Note:**
> При установке из Zabbix Git репозитория, Вам необходимо выполнить следующую команду прежде, чем приступать к последующим шагам:

 `make dbschema`

MySQL/MariaDB

Набор символов utf8mb4 с сопоставлением utf8mb4_bin поддерживается для корректной работы сервера/прокси Zabbix с базами данных MySQL/MariaDB.
 Чтобы избежать проблем в будущем, настоятельно рекомендуется использовать набор символов utf8mb4 с сопоставлением utf8mb4_bin как для новых, так и для существующих установок. Чтобы преобразовать базы данных, использующие набор символов utf8mb3 и сопоставление utf8mb3_bin, в набор символов utf8mb4 и сопоставление utf8mb4_bin, следуйте инструкциям в разделе Восстановление набора символов и сопоставления базы данных Zabbix.

Детерминированные триггеры необходимо создавать во время импорта схемы. В MySQL и MariaDB для этого требуется установить `GLOBAL log_bin_trust_function_creators = 1`, если включено двоичное журналирование, отсутствуют привилегии суперпользователя и в файле конфигурации MySQL не установлено значение `log_bin_trust_function_creators = 1`.

Если вы устанавливаете Zabbix из пакетов или контейнеров, см. соответствующие инструкции.

Если вы устанавливаете Zabbix из исходного кода:

- Создайте и настройте базу данных и пользователя. В целях безопасности используйте надежный уникальный пароль для пользователя базы данных Zabbix.

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
```

```
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- Импортируйте данные в базу данных и установите набор символов utf8mb4 в качестве набора символов по умолчанию (предполагается, что вы находитесь в корневом каталоге исходного кода Zabbix). Для базы данных прокси Zabbix следует импортировать только `schema.sql` (без `images.sql` и `data.sql`).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
####           ,                          Zabbix
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` можно отключить после успешного импорта схемы:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Вам необходимо иметь пользователя базы данных с правами на создание объектов базы данных.

Если вы устанавливаете Zabbix из пакетов или контейнеров, см. соответствующие инструкции.

Если вы устанавливаете Zabbix из исходных кодов:

- Создайте пользователя базы данных.

Следующая команда shell создаст пользователя `zabbix`. Укажите пароль при запросе и повторите пароль (обратите внимание, сначала у вас могут запросить пароль sudo):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Создайте базу данных.

Следующая команда shell создаст базу данных `zabbix` (последний параметр) с ранее созданным пользователем в качестве владельца (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Импортируйте начальную схему и данные (предполагается, что вы находитесь в корневом каталоге исходных кодов Zabbix). Для базы данных Zabbix proxy следует импортировать только `schema.sql` (без `images.sql` и `data.sql`).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### stop here if you are creating database for Zabbix proxy
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Приведенные выше команды даны в качестве примера, который будет работать в большинстве установок GNU/Linux. Вы можете использовать другие команды в зависимости от того, как настроена ваша система/база данных, например:

 `psql -U <username>`

 Если у вас возникнут трудности при настройке базы данных, обратитесь к администратору базы данных.

TimescaleDB

Инструкции по созданию и настройке TimescaleDB приведены в отдельном разделе.

Elasticsearch и ClickHouse

Zabbix также может хранить данные истории в Elasticsearch или ClickHouse. Инструкции по настройке см. на страницах по ссылкам.

SQLite

Использование SQLite поддерживается только для **прокси Zabbix**!

База данных будет создана автоматически, если она не существует.

Вернуться к разделу установки.

Исправление набора символов и сопоставления базы данных Zabbix

MySQL/MariaDB

Исторически MySQL и его производные использовали 'utf8' как псевдоним для utf8mb3 - собственной 3-байтной реализации стандарта UTF8, который является 4-байтным. Начиная с MySQL 8.0.28 и MariaDB 10.6.1, набор символов 'utf8mb3' считается устаревшим, и в какой-то момент его поддержка будет удалена, а 'utf8' станет ссылкой на 'utf8mb4'. Начиная с Zabbix 6.0, поддерживается 'utf8mb4'. Чтобы избежать будущих проблем, настоятельно рекомендуется использовать 'utf8mb4'. Еще одно преимущество перехода на 'utf8mb4' - поддержка дополнительных символов Unicode.

Warning:
Поскольку версии до Zabbix 6.0 не поддерживают utf8mb4, перед выполнением преобразования utf8mb4 обязательно сначала обновите сервер Zabbix и схему БД до 6.0.x или более поздней версии.

1. Проверьте набор символов и сопоставление базы данных.

Например:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

Или:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Как видим, набор символов здесь не 'utf8mb4', а сопоставление не 'utf8mb4_bin', поэтому их нужно исправить.

2. Остановите Zabbix.

3. Создайте резервную копию базы данных!

4. Исправьте набор символов и сопоставление на уровне базы данных:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Исправленные значения:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Загрузите скрипт для исправления набора символов и сопоставления на уровне таблиц и столбцов:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Выполните скрипт:

```
                SET @ZABBIX_DATABASE = '<your DB name>';
   MariaDB →    set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
   MariaDB →    set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

Обратите внимание, что 'utf8mb4' может занимать немного больше места на диске.

7. Если ошибок нет - вы можете создать резервную копию базы данных с исправленной базой данных.

8. Запустите Zabbix.

Обновление базы данных до первичных ключей

Обзор

В этом разделе приведены инструкции по ручному переводу существующих установок на первичные ключи в таблицах.

Переход на первичные ключи оптимизирует индексирование и доступ к данным, что может ускорить запросы и сэкономить место. Это также улучшает управление данными и синхронизацию в кластерных конфигурациях, помогая масштабированию и обеспечивая надежную работу системы даже при отказе некоторых серверов.

Attention:
Инструкции, приведенные на этой странице, предназначены для опытных пользователей и могут потребовать адаптации под вашу конкретную конфигурацию. Переход на первичные ключи может быть длительным и ресурсоемким. Убедитесь, что доступно достаточно свободного места на диске; в зависимости от размера базы данных и хранимых данных процесс может потребовать до 2,5 раза больше места, чем сейчас занимают таблицы истории.

Первичные ключи используются для всех таблиц в новых установках начиная с Zabbix 6.0.

Автоматического обновления базы данных до первичных ключей нет; однако существующие установки можно обновить вручную **после** обновления Zabbix server до 6.0 или более новой версии.

Attention:
Начиная с Zabbix 7.0, при обновлении таблиц до первичных ключей таблицы также переводятся на типы данных double precision.

 Если у вас Zabbix 7.0 (или более поздняя версия), таблицы уже используют double precision. Однако инструкции на этой странице по-прежнему можно использовать для обновления таблиц до первичных ключей без влияния на таблицы, которые уже используют double precision.

 Если у вас Zabbix 6.4 (или более ранняя версия), рекомендуется сначала обновить таблицы до double precision. Дополнительную информацию см. в разделе Обновление до числовых значений расширенного диапазона в документации Zabbix 7.0.

Доступны инструкции для:

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

Важные примечания

Чтобы выполнить обновление базы данных:

1. Остановите сервер Zabbix.

Настоятельно рекомендуется останавливать сервер Zabbix на время обновления. Однако, если это абсолютно необходимо, вы можете выполнить обновление, пока сервер работает (только для MySQL, MariaDB и PostgreSQL без TimescaleDB).

2. Создайте резервную копию базы данных.
3. Установите последнюю версию пакета zabbix-sql-scripts, совместимую с вашей версией Zabbix (например, для RHEL: `dnf install zabbix-sql-scripts`).
4. Запустите скрипты для вашей базы данных.
5. Запустите сервер Zabbix.

Warning:
Запускайте скрипты только для базы данных сервера. Прокси не получит никаких преимуществ от этого обновления.

Если база данных использует разделы, обратитесь за помощью к администратору БД или в службу поддержки Zabbix.

CSV-файлы можно удалить после успешного обновления до первичных ключей.

При необходимости веб-интерфейс Zabbix можно перевести в режим обслуживания.

MariaDB/MySQL

Экспорт и импорт необходимо выполнять в tmux/screen, чтобы сеанс не был разорван.

См. также: Important notes

MySQL 8.4+ с mysqlsh

Этот метод можно использовать при работающем сервере Zabbix, но рекомендуется остановить сервер на время обновления. MySQL Shell (*mysqlsh*) должен быть установлен и иметь возможность подключаться к БД.

- Войдите в консоль MySQL как root (рекомендуется) или как любой пользователь с привилегиями FILE.
- Запустите MySQL с включенной переменной local_infile.
- Переименуйте старые таблицы и создайте новые, выполнив `history_upgrade_prepare.sql`.

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Экспортируйте и импортируйте данные.

Подключитесь через mysqlsh. При использовании подключения через сокет может потребоваться указать путь.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Переключитесь в режим JavaScript с помощью:

```
\js
```

Затем выполните приведенный ниже код (при необходимости CSVPATH можно изменить):

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

Если вы получите сообщение "JavaScript is not supported", значит в вашей установке MySQL Shell отсутствует поддержка JS. В этом случае установите официальный пакет MySQL Shell от Oracle (или соберите его из исходного кода), чтобы был доступен режим JavaScript.

- Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

MariaDB/MySQL 8.4+ без mysqlsh

Этот способ обновления занимает больше времени и должен использоваться только в том случае, если обновление с помощью *mysqlsh* невозможно.

Обновление таблиц

- Войдите в консоль MySQL как root (рекомендуется) или любой пользователь с привилегиями FILE.
- Если выполняется миграция при работающем сервере Zabbix, запустите MySQL с включенной переменной local_infile.
- Переименуйте старые таблицы и создайте новые, выполнив `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Миграция при остановленном сервере

`max_execution_time` (в MySQL) или `max_statement_time` (в MariaDB) должны быть отключены перед миграцией данных, чтобы избежать тайм-аута во время миграции.

Для MySQL:

```
SET @@max_execution_time=0;
```

Для MariaDB:

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

Миграция при работающем сервере

Проверьте, для каких путей включен импорт/экспорт:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Если значение *secure_file_priv* является путем к каталогу, экспорт/импорт будет выполняться для файлов в этом каталоге. В этом случае соответствующим образом измените пути к файлам в запросах или установите значение *secure_file_priv* в пустую строку на время обновления.

Если значение *secure_file_priv* пустое, экспорт/импорт можно выполнять из любого расположения.

Если значение *secure_file_priv* равно NULL, задайте путь к каталогу, содержащему экспортированные данные таблиц (в примере выше это '/var/lib/mysql-files/').

Дополнительные сведения см. в документации MySQL или документации MariaDB.

`max_execution_time` (в MySQL) или `max_statement_time` (в MariaDB) должны быть отключены перед экспортом данных, чтобы избежать тайм-аута во время экспорта.

Для MySQL:

```
SET @@max_execution_time=0;
```

Для MariaDB:

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

PostgreSQL

Экспорт и импорт необходимо выполнять в tmux/screen, чтобы сеанс не был разорван. Для установок с TimescaleDB пропустите этот раздел и перейдите к PostgreSQL + TimescaleDB.

См. также: Важные примечания

Обновление таблиц

- Переименуйте таблицы с помощью `history_upgrade_prepare.sql`:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Миграция с остановленным сервером

- Экспортируйте текущую историю, импортируйте ее во временную таблицу, затем вставьте данные в новые таблицы, игнорируя дубликаты:

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

См. советы по повышению производительности INSERT: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

Миграция при работающем сервере

- Сделайте экспорт текущей истории, импорт её во временную таблицу, затем вставку данных в новые таблицы, игнорируя дубликаты:

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    value                   DOUBLE PRECISION DEFAULT '0.0000'        NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    value                   numeric(20)     DEFAULT '0'             NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_str (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    value                   varchar(255)    DEFAULT ''              NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,n

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    timestamp               integer         DEFAULT '0'             NOT NULL,
    source                  varchar(64)     DEFAULT ''              NOT NULL,
    severity                integer         DEFAULT '0'             NOT NULL,
    value                   text            DEFAULT ''              NOT NULL,
    logeventid              integer         DEFAULT '0'             NOT NULL,
```

```
    ns                      integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid                  bigint                                    NOT NULL,
    clock                   integer         DEFAULT '0'               NOT NULL,
    value                   text            DEFAULT ''                NOT NULL,
    ns                      integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- Следуйте инструкциям по пост-миграции для удаления старых таблиц.

PostgreSQL + TimescaleDB

Экспорт и импорт должны выполняться в tmux/screen, чтобы сеанс не был разорван. Во время обновления сервер Zabbix должен быть остановлен.

См. также: Important notes

- Переименуйте таблицы с помощью `history_upgrade_prepare.sql`.
  - Если сжатие включено (в стандартной установке), выполните скрипт из /usr/share/zabbix/sql-scripts/postgresql

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - Если сжатие отключено, выполните скрипт из /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- Запустите скрипты миграции hypertable TimescaleDB в зависимости от настроек сжатия:
  - Если сжатие включено (в стандартной установке), выполните скрипты из /usr/share/zabbix/sql-scripts/postgresc

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
    ```

  - Если сжатие отключено, выполните скрипты из /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
    ```

См. также: Tips для повышения производительности INSERT.

- Следуйте post-migration instructions, чтобы удалить старые таблицы.

Пост-миграция

Для всех баз данных после завершения миграции сделайте следующее:

- Проверьте, что всё работает, как ожидается.
- Удалите старые таблицы:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- Для TimescaleDB также удалите следующую старую таблицу:

```
DROP TABLE trends_old;
```

Смотрите также

- Подготовка таблицы журнала аудита к партиционированию

Подготовка таблицы auditlog к секционированию

Обзор

Некоторые базы данных (например, MySQL) требуют, чтобы столбец партиционирования был частью уникального ограничения таблицы (table's unique constraint). Поэтому для партиционирования таблицы `auditlog` по времени первичный ключ необходимо изменить с `auditid` на составной ключ `auditid` + `clock`.

В этом разделе приведены инструкции по изменению первичного ключа таблицы `auditlog`.

Attention:
Инструкции, представленные на этой странице, предназначены для опытных пользователей. Обратите внимание, что эти инструкции, возможно, придется скорректировать для вашей конкретной конфигурации. Изменение первичного ключа также может быть несовместимо с будущими обновлениями, поэтому может потребоваться ручная обработка будущих обновлений.

 Изменение первичного ключа может быть ресурсоёмкой операцией, которая занимает много времени в зависимости от размера таблицы `auditlog`. Рекомендуется остановить сервер Zabbix и переключить веб-интерфейс Zabbix в режим обслуживания на время изменения. Однако, если это абсолютно необходимо, есть способ изменить первичный ключ без простоя (смотрите ниже).

Партиционирование таблицы `auditlog` может улучшить, например, процедуру очистки истории в больших инсталляциях.

Хотя очистка истории Zabbix в настоящее время не может использовать преимущества партиционированных таблиц (за исключением TimescaleDB), вы можете отключить очистку истории Zabbix и удалять партиции с помощью скриптов.

Начиная с версии Zabbix 7.0, таблица `auditlog` для TimescaleDB была преобразована в гипертаблицу, что позволяет администратору удалять данные по частям. Чтобы обновить существующую таблицу `auditlog` до гипертаблицы, смотрите Обновление схемы TimescaleDB.

MySQL

Важные замечания по перестроению индексов

MySQL автоматически перестраивает индексы для первичного ключа во время операции `ALTER TABLE`. Однако, чтобы обеспечить оптимальную производительность базы данных, настоятельно рекомендуется также вручную перестроить индексы с помощью оператора `OPTIMIZE TABLE`.

Перестроение индексов может временно потребовать столько же дополнительного дискового пространства, сколько использует сама таблица. Чтобы получить текущий размер данных и индексов, можно выполнить следующие операторы:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Если доступное дисковое пространство вызывает беспокойство, следуйте инструкциям Изменение первичного ключа без простоя. Также доступны другие варианты:

- Увеличение параметра MySQL `sort_buffer_size` [en] может помочь сократить использование дискового пространства при ручном перестроении индексов. Однако изменение этой переменной может повлиять на общее использование памяти базы данных.
- Рассмотрите возможность освобождения места путем удаления потенциально ненужных данных.
- Рассмотрите возможность уменьшения параметра очистки истории *Период хранения данных (Data storage period)* перед выполнением очистки истории.

Изменение первичного ключа с простоем

1. Удалите текущий первичный ключ таблицы `auditlog` и добавьте новый первичный ключ.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Перестройте индексы (необязательно, но настоятельно рекомендуется, смотрите Важные примечания по перестроению индексов).

```
OPTIMIZE TABLE auditlog;
```

Изменение первичного ключа без простоя

Здесь описан ручной метод изменения первичного ключа. В качестве альтернативы вы можете использовать набор инструментов pt-online-schema-change [en] от Percona. Этот набор инструментов автоматически выполняет следующие действия, а также минимизирует пространство, используемое для изменения таблицы auditlog.

1. Создайте новую таблицу с новым первичным ключом и создайте индексы.

```
CREATE TABLE `auditlog_new` (
  `auditid`            varchar(25)                          NOT NULL,
  `userid`             bigint unsigned                      NULL,
  `username`           varchar(100)    DEFAULT ''           NOT NULL,
  `clock`              integer         DEFAULT '0'          NOT NULL,
  `ip`                 varchar(39)     DEFAULT ''           NOT NULL,
  `action`             integer         DEFAULT '0'          NOT NULL,
  `resourcetype`       integer         DEFAULT '0'          NOT NULL,
  `resourceid`         bigint unsigned                      NULL,
  `resource_cuid`      varchar(25)                          NULL,
  `resourcename`       varchar(255)    DEFAULT ''           NOT NULL,
  `recordsetid`        varchar(25)                          NOT NULL,
  `details`            longtext                             NOT NULL,
  PRIMARY KEY (auditid,clock)
) ENGINE=InnoDB;
CREATE INDEX `auditlog_1` ON `auditlog_new` (`userid`,`clock`);
CREATE INDEX `auditlog_2` ON `auditlog_new` (`clock`);
CREATE INDEX `auditlog_3` ON `auditlog_new` (`resourcetype`,`resourceid`);
```

2. Поменяйте таблицы местами.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Скопируйте данные из старой таблицы в новую.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Это можно сделать частями (несколько операторов INSERT INTO с условиями WHERE clock по мере необходимости), чтобы избежать чрезмерного использования ресурсов.

4. Удалите старую таблицу.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Важные замечания по перестроению индексов

PostgreSQL автоматически перестраивает индексы для первичного ключа во время операции ALTER TABLE. Однако, чтобы обеспечить оптимальную производительность базы данных, настоятельно рекомендуется также вручную перестроить индексы с помощью оператора REINDEX TABLE CONCURRENTLY.

Перестроение индексов может временно потребовать до трёхкратного дискового пространства, используемого индексами в данный момент. Чтобы получить текущий размер индексов, можно выполнить следующий запрос:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Если доступное дисковое пространство вызывает беспокойство, следуйте инструкциям Изменение первичного ключа без простоя. Также доступны другие варианты:

- Увеличение параметра PostgreSQL maintenance_work_mem [en] может помочь сократить использование дискового пространства при ручном перестроении индексов. Однако, изменение этой переменной может повлиять на общее использование памяти базы данных.
- Если у вас есть другой диск или табличное пространство с большим свободным местом, вы можете рассмотреть возможность изменения временного места хранения для перестроения индекса. Вы можете задать параметр PostgreSQL temp_tablespaces [en], чтобы указать другое табличное пространство для временных объектов.
- Рассмотрите возможность освобождения места путём удаления потенциально ненужных данных.
- Рассмотрите возможность уменьшения параметра очистки истории *Период хранения данных (Data storage period)* перед выполнением очистки истории.

Изменение первичного ключа с простоем

1. Удалите текущий первичный ключ таблицы auditlog и добавьте новый первичный ключ.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Перестройте индексы (необязательно, но настоятельно рекомендуется, смотрите Важные примечания по перестроению индексов).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Изменение первичного ключа без простоя

Здесь описан ручной метод изменения первичного ключа. В качестве альтернативы можно рассмотреть расширение pg_repack для создания новой таблицы, копирования данных и замены таблиц.

1. Создайте новую таблицу с новым первичным ключом и создайте индексы.

```
CREATE TABLE auditlog_new (
  auditid              varchar(25)                         NOT NULL,
  userid               bigint                              NULL,
  username             varchar(100)    DEFAULT ''          NOT NULL,
  clock                integer         DEFAULT '0'         NOT NULL,
  ip                   varchar(39)     DEFAULT ''          NOT NULL,
  action               integer         DEFAULT '0'         NOT NULL,
  resourcetype         integer         DEFAULT '0'         NOT NULL,
  resourceid           bigint                              NULL,
  resource_cuid        varchar(25)                         NULL,
  resourcename         varchar(255)    DEFAULT ''          NOT NULL,
  recordsetid          varchar(25)                         NOT NULL,
  details              text            DEFAULT ''          NOT NULL,
  PRIMARY KEY (auditid,clock)
);
CREATE INDEX auditlog_new_1 ON auditlog_new (userid,clock);
CREATE INDEX auditlog_new_2 ON auditlog_new (clock);
CREATE INDEX auditlog_new_3 ON auditlog_new (resourcetype,resourceid);
```

2. Поменяйте таблицы местами.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Скопируйте данные из старой таблицы в новую.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Это можно сделать частями (несколько операторов INSERT INTO с условиями WHERE clock по мере необходимости), чтобы избежать чрезмерного использования ресурсов.

4. Удалите старую таблицу.

```
DROP TABLE auditlog_old;
```

Смотрите также

- Обновление базы данных для использования первичных ключей

Настройка TimescaleDB

Обзор

Для обеспечения большего уровня производительности при масштабировании, Zabbix поддерживает TimescaleDB — решение на основе базы данных PostgreSQL для автоматического партиционирования данных на фрагменты (chunks) на основе времени.

> **Warning:**
> В настоящее время TimescaleDB не поддерживается Zabbix прокси-сервером.

Инструкции на этой странице могут быть использованы для следующих сценариев:

- Создание базы данных TimescaleDB или для миграции с существующих таблиц PostgreSQL на TimescaleDB (смотрите Настройка).

- Обновление существующей схемы базы данных TimescaleDB при обновлении Zabbix (смотрите Обновление схемы TimescaleDB).

Конфигурация

Предварительные требования: расширение TimescaleDB поддерживаемой версии, установленное на сервере базы данных. Инструкции по установке см. в документации TimescaleDB.

> **Warning:**
> Перед установкой TimescaleDB установите поддерживаемый выпуск PostgreSQL из официального репозитория PostgreSQL.

Включите расширение TimescaleDB для конкретной базы данных, выполнив:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

Для выполнения этой команды требуются права администратора базы данных.

> **Note:**
> Если вы используете схему базы данных, отличную от 'public', необходимо добавить в приведенную выше команду предложение SCHEMA. Например:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Затем выполните скрипт `postgresql/timescaledb/schema.sql`. Для новых установок скрипт необходимо запускать после того, как обычная база данных PostgreSQL будет создана с начальной схемой/данными (см. создание базы данных).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Пожалуйста, игнорируйте предупреждения о том, что при выполнении скрипта `schema.sql` в TimescaleDB версии 2.9.0 и выше не соблюдаются рекомендации по лучшим практикам. Несмотря на это предупреждение, настройка будет успешно завершена.

Миграция существующих данных history, trends и audit log может занять много времени. На период миграции сервер Zabbix и веб-интерфейс должны быть остановлены.

Скрипт `schema.sql` задает следующие параметры housekeeping:

- Переопределить период history элемента данных
- Переопределить период trends элемента данных

Чтобы использовать разбиение housekeeping на разделы для history и trends, необходимо включить оба этих параметра. Также можно включить переопределение отдельно только для history или только для trends.

Для PostgreSQL и TimescaleDB скрипт `postgresql/timescaledb/schema.sql` задает два дополнительных параметра:

- Включить сжатие
- Сжимать записи старше 7 дней

Для успешного удаления сжатых данных процессом очистки истории, обе опции *Переопределить период хранения истории элементов данных (Override item history period)* и *Переопределить период хранения динамики изменения элементов данных (Override item trend period)* должны быть активированы. Если переопределение отключено, а таблицы имеют сжатые фрагменты, процесс очистки истории не удалит данные из этих таблиц, а в разделах *Очистка истории (Housekeeping)* и *Информация о системе (System information)* будет отображено предупреждение о некорректной конфигурации.

Все эти параметры можно изменить в разделе *Администрирование* → *Очистка истории* (*Administration* → *Housekeeping*) после установки.

> **Note:**
> Возможно, вы захотите запустить инструмент timescaledb-tune, предоставляемый TimescaleDB, чтобы оптимизировать параметры конфигурации PostgreSQL в вашем файле `postgresql.conf`.

Обновление схемы TimescaleDB

При обновлении Zabbix до версии, содержащей новые гипертаблицы TimescaleDB, сервер Zabbix не настраивает эти гипертаблицы автоматически (например, при обновлении с Zabbix 6.4 до 8.0, так как в версиях 7.0.0 и 7.0.2 были добавлены новые гипертаблицы).

Чтобы настроить новые гипертаблицы TimescaleDB, выполните следующие шаги:

1. Запустите сервер Zabbix; это обновит существующую базу данных.
2. Убедитесь по файлу журнала сервера, что обновление базы данных завершено; после завершения остановите сервер Zabbix. Обратите внимание, что сервер записывает предупреждение, если пытается включить сжатие для таблицы, которая не является гипертаблицей.
3. Запустите скрипт `postgresql/timescaledb/schema.sql`; он настроит новые гипертаблицы TimescaleDB. Обратите внимание, что начиная с Zabbix 7.0.0 расположение и имя скрипта были изменены с `postgresql/timescaledb.sql` на `postgresql/timescaledb/schema.sql`.

Attention:
Пожалуйста, игнорируйте предупреждающие сообщения о несоблюдении рекомендаций по лучшим практикам при запуске скрипта `schema.sql` на TimescaleDB версии 2.9.0 и выше. Несмотря на это предупреждение, настройка будет успешно завершена.

Сжатие TimescaleDB

Нативное сжатие TimescaleDB поддерживается для всех таблиц Zabbix, которые являются hypertable в TimescaleDB. Во время обновления или миграции на TimescaleDB начальное сжатие больших таблиц может занять много времени.

Обратите внимание, что сжатие поддерживается по лицензии сообщества Timescale "timescale" и не поддерживается по лицензии Apache 2.0 "apache". Если Zabbix обнаружит, что сжатие не поддерживается, в журнал сервера Zabbix будет записано предупреждение, а пользователи не смогут включить сжатие в веб-интерфейсе.

Note:
Перед использованием сжатия пользователям рекомендуется ознакомиться с разделом о сжатии в документации TimescaleDB.

Обратите внимание, что сжатие накладывает определенные ограничения, в частности:

- Изменение сжатых чанков (вставка, удаление, обновление) не допускается
- Изменение схемы для сжатых таблиц не допускается.

Параметры сжатия можно изменить в блоке *Сжатие истории, трендов и журнала аудита* в разделе *Администрирование > Обслуживание* веб-интерфейса Zabbix.

Параметр	По умолчанию	Комментарии
Включить сжатие	Включено	Установка или снятие флажка не активирует и не деактивирует сжатие немедленно. Поскольку сжатием управляет Housekeeper, изменения вступят в силу максимум через 2 раза `HousekeepingFrequency` часов (задается в `zabbix_server.conf`) После отключения сжатия новые чанки, попадающие в период сжатия, сжиматься не будут. Однако все ранее сжатые данные останутся сжатыми. Чтобы распаковать ранее сжатые чанки, следуйте инструкциям в документации TimescaleDB.
Сжимать записи старше чем	7d	При обновлении с более старых версий Zabbix с поддержкой TimescaleDB сжатие по умолчанию включено не будет. Этот параметр не может быть меньше 7 дней. Из-за неизменяемости сжатых чанков все поздние данные (например, данные, задержанные прокси) старше этого значения будут отброшены.

Для лучшей производительности обновления трендов вы можете уменьшить значение "chunk_time_interval" для таблиц `trends` и `trends_uint` с 30 дней до 7 дней или меньше, в зависимости от того, сколько элементов данных используют тренды. Цель этой настройки — соответствовать рекомендациям TimescaleDB и обеспечить, чтобы размер чанка оставался в пределах доступных системных ресурсов.

Настройка Elasticsearch

Zabbix может хранить данные истории в Elasticsearch в качестве альтернативы реляционной базе данных.

Attention:
Поддержка Elasticsearch в настоящее время является экспериментальной.

В этом руководстве описана настройка поддерживаемых версий Elasticsearch. Если вы используете другую версию, некоторые функции могут работать не так, как ожидается.

Elasticsearch может хранить следующие типы значений:

Тип значения элемента данных	Таблица базы данных	Тип Elasticsearch
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text
Binary	history_bin	не поддерживается Zabbix
JSON	history_json	json

Note:
Elasticsearch не принимает массивы JSON. Значение JSON должно быть либо одним объектом, либо набором объектов.

Важные примечания

- Elasticsearch требует libcurl. Подробности см. в разделе требования.
- housekeeper не удаляет данные из Elasticsearch. Чтобы управлять сроком хранения данных, см. Index Lifecycle Management.
- Zabbix **не** вычисляет и не хранит тренды в Elasticsearch. Рассмотрите возможность увеличения периода хранения истории, чтобы сохранить более старые данные.
- При использовании Elasticsearch запросы диапазона, извлекающие значения из базы данных, ограничены временной меткой периода хранения данных.
- Elasticsearch не поддерживается для прокси Zabbix.

Если Elasticsearch еще не установлен, перед продолжением обратитесь к официальному руководству по установке.

Настройка Elasticsearch

Чтобы хранить исторические данные в Elasticsearch, необходимо:

- Создать индекс для каждого типа значений, который вы хотите хранить — именно здесь Elasticsearch хранит данные, аналогично таблице в реляционной базе данных.
- Определить mapping для каждого индекса — он задаёт структуру данных, аналогично схеме таблицы.
- Настроить ingest pipeline для обработки значений перед сохранением (требуется для значений JSON и индексов на основе даты).

Elasticsearch может хранить данные в одном индексе для каждого типа значений или в нескольких индексах на основе даты. Оба подхода описаны ниже.

Хранение истории в одном индексе

При таком подходе все исторические данные для заданного типа значений записываются в один индекс (например, `uint` или `text`).

Чтобы создать индекс для типа значений *Numeric (unsigned)*, отправьте следующий запрос (с `/uint` в URL) вашему экземпляру Elasticsearch:

```
curl -X PUT \
 http://localhost:9200/uint \
 -H 'content-type:application/json' \
 -d '{
     "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
```

```
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": { "type": "long" }
      }
   }
}'
```

Elasticsearch ответит подтверждением о том, что индекс был создан:

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Аналогичные запросы необходимо отправить для каждого дополнительного типа значений, который вы хотите хранить в Elasticsearch.

> **Note:**
> Сопоставления для всех типов значений доступны в репозитории исходного кода Zabbix.

Например, чтобы создать индекс для типа значений *Text*:

```
curl -X PUT \
 http://localhost:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Тип значения JSON

В отличие от других типов значений, значения JSON требуют дополнительной обработки перед сохранением.

Приведённый ниже индекс использует отдельные поля для разобранных и необработанных значений, поэтому для разбора каждого значения как JSON и сохранения его в правильном поле необходим ingest pipeline.

Чтобы создать индекс для типа значения *JSON*, отправьте следующий запрос (с `/json` в URL) вашему экземпляру Elasticsearch.

```
curl -X PUT \
 http://localhost:9200/json \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
```

```
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
    }
  }
}'
```

Затем создайте ingest pipeline:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json \
 -H 'content-type:application/json' \
 -d '{
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "set": {
            "if": "ctx.value_parsed == null",
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ],
   "on_failure": [
      {
         "set": {
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ]
}'
```

Elasticsearch ответит подтверждением того, что ingest pipeline был создан:

```
{"acknowledged": true}
```

Хранение истории в индексах на основе даты

Вместо записи всех данных истории в один индекс (например, `uint`) Elasticsearch может распределять эти данные по нескольким индексам на основе даты (например, `uint-2026-01-01`, `uint-2026-01-02`). Это упрощает управление объемом данных и сроками хранения с течением времени.

Чтобы включить это, необходимо:

- Создать шаблон индекса для каждого типа значения, который вы хотите хранить — это сообщает Elasticsearch, какое сопоставление применять при автоматическом создании нового индекса на основе даты.
- Создать конвейер ingest для каждого типа значения — он обрабатывает каждое входящее значение и направляет его в правильный индекс на основе даты.

Шаблоны индексов

Чтобы создать шаблон для индекса `text`, отправьте запрос со следующими данными:

- Используйте `_template/text_template` в URL вашего экземпляра Elasticsearch.
- Используйте `"text*"` в поле `"index_patterns"` для сопоставления имени индекса.
- Используйте mapping для типа значения `text` (см. mappings в репозитории исходного кода Zabbix).

```
curl -X PUT \
 http://localhost:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "text*" ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Шаблон для индекса `json`:

```
curl -X PUT \
 http://localhost:9200/_template/json_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "json*" ],
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
      }
   }
}'
```

Конвейеры ingest

Чтобы создать конвейер ingest для индекса `text`:

- Используйте `_ingest/pipeline/text-pipeline` в URL вашего экземпляра Elasticsearch.
- Добавьте процессор `date_index_name`, чтобы направлять каждое значение в правильный индекс на основе даты в соответствии с его временной меткой.

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/text-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily text index naming",
   "processors": [
      {
         "date_index_name": {
```

```
            "field": "clock",
            "date_formats": ["UNIX"],
            "index_name_prefix": "text-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Для индекса `json` конвейер также должен разбирать значение JSON перед его направлением в правильный индекс:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily json index naming"
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "script": {
            "source": "if (ctx.value_parsed == null || !(ctx.value_parsed instanceof Map)) { ctx.value_raw
         }
      },
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [ "UNIX" ],
            "index_name_prefix": "json-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Настройка сервера Zabbix

В файле конфигурации сервера Zabbix (`zabbix_server.conf`) задайте параметр `HistoryProvider`.

Например, чтобы хранить значения типов *Character*, *Log*, *Text* и *JSON* в Elasticsearch (при этом значения *Numeric* оставлять в базе данных):

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Если вы используете индексы на основе даты, добавьте `date_index=1` к параметру:

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

После внесения изменений перезапустите сервер Zabbix:

```
systemctl restart zabbix-server
```

Настройка веб-интерфейса Zabbix

В вашем файле конфигурации веб-интерфейса Zabbix (`zabbix.conf.php`) задайте переменную `$HISTORY_PROVIDERS` в соответствии с конфигурацией сервера:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Устранение неполадок

Следующие шаги могут помочь вам устранить проблемы с настройкой Elasticsearch:

1. Проверьте журналы Elasticsearch или сервера Zabbix на наличие ошибок.
2. Чтобы выявить медленные запросы, используйте параметр `log_slow_queries` в параметре конфигурации сервера Zabbix `HistoryProvider`.
3. Убедитесь, что Elasticsearch разрешает доступ с сервера Zabbix и веб-интерфейса Zabbix.
4. Убедитесь, что `auto_create_index` включен:

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Чтобы включить его, отправьте следующий запрос:

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. Проверьте, что mappings, шаблоны и ingest pipelines настроены правильно, отправив запросы `GET` к соответствующим URL:

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

Вы можете сравнить полученные ответы с ожидаемыми ответами в документации API Elasticsearch.

6. Проверьте, не находятся ли какие-либо shards в состоянии сбоя; перезапуск Elasticsearch может решить эту проблему.
7. Выполните запрос к Elasticsearch, чтобы проверить, сохраняются ли данные, собранные Zabbix, например:

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Если вам нужно сбросить настройку Elasticsearch и начать заново, вы можете удалить все индексы, шаблоны и ingest pipelines:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

Настройка ClickHouse

Zabbix может хранить данные истории в ClickHouse в качестве альтернативы реляционной базе данных.

Note:
ClickHouse также можно использовать как источник телеметрических данных для мониторинга производительности приложений (APM); подробную информацию см. в разделе *Администрирование* > *Источник данных*.

В этом руководстве описана настройка поддерживаемых версий ClickHouse. При использовании другой версии некоторые функции могут работать некорректно.

ClickHouse может хранить следующие типы значений:

Тип значения элемента данных	Таблица базы данных	Тип ClickHouse
Числовой (беззнаковый)	history_uint	uint
Числовой (с плавающей точкой)	history	dbl
Символьный	history_str	str
Журнал	history_log	log
Текстовый	history_text	text
Двоичный	history_bin	не поддерживается Zabbix
JSON	history_json	json

Note:
ClickHouse не принимает массивы JSON. Значение JSON должно быть либо отдельным объектом, либо набором объектов. Кроме того, ClickHouse обрабатывает ключи JSON со значением NULL так же, как отсутствующие ключи.

Важные примечания

- Сборщик устаревших данных не удаляет данные из ClickHouse. Чтобы управлять сроком хранения данных, настройте период хранения данных ClickHouse в ClickHouse.
- Zabbix **не** вычисляет и не хранит тренды в ClickHouse. Рассмотрите возможность увеличения периода хранения истории, чтобы сохранять более старые данные.
- Если вы хотите перенести данные истории из существующей базы данных Zabbix (MySQL или PostgreSQL) в ClickHouse, см. скрипты миграции схемы ClickHouse и истории.
- ClickHouse не поддерживается в качестве хранилища истории для прокси Zabbix, но может использоваться как источник данных телеметрии для прокси Zabbix с включенным APM.
- Если у вас есть кластер ClickHouse, см. Настройка кластеров ClickHouse.

Настройка ClickHouse

Вам нужно создать и настроить базу данных Zabbix и пользователя, а также импортировать схему базы данных.

В этом руководстве приведены инструкции для установок ClickHouse через Docker или пакеты.

Docker

1. Создайте и настройте базу данных Zabbix и пользователя при запуске контейнера ClickHouse:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Убедитесь, что ClickHouse работает и вы можете подключиться к нему:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"
```

```
#### 26.4.4.38
```

3. Импортируйте схему базы данных с помощью скрипта history_all.sh из каталога Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
Период хранения данных (Time-To-Live, или TTL) для ClickHouse по умолчанию составляет 31 день. Чтобы изменить его, используйте параметр `--ttl` при импорте схемы базы данных (например, `--ttl 604800` для 7 дней) или настройте его позже.

4. Проверьте, что таблицы были созданы:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Пакеты

1. Запустите ClickHouse:

```
sudo service clickhouse-server start
```

2. Создайте и настройте базу данных и пользователя Zabbix:

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Импортируйте схему базы данных с помощью скрипта history_all.sh из каталога Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
Период хранения данных (Time-To-Live, или TTL) для ClickHouse по умолчанию составляет 31 день. Чтобы изменить его, используйте параметр `--ttl` при импорте схемы базы данных (например, `--ttl 604800` для 7 дней) или настройте его позже.

4. Убедитесь, что таблицы были созданы:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
```

```
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Настройка сервера Zabbix

В файле конфигурации сервера Zabbix (`zabbix_server.conf`) задайте параметр `HistoryProvider`.

Например, чтобы хранить все поддерживаемые типы значений в ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

После внесения изменений перезапустите сервер Zabbix:

```
systemctl restart zabbix-server
```

Настройка веб-интерфейса Zabbix

В файле конфигурации веб-интерфейса Zabbix (`zabbix.conf.php`) задайте переменную `$HISTORY_PROVIDERS` в соответствии с конфигурацией сервера:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Дополнительная настройка

Шаги ниже необязательны. Они не нужны для базовой настройки.

Настройка периода хранения данных ClickHouse

Период хранения данных (Time-To-Live, или TTL) для ClickHouse по умолчанию составляет 31 день. Чтобы изменить его, выполните команды ниже.

В примерах ниже используется Docker. Если вы установили ClickHouse с помощью пакетов, выполняйте запросы напрямую в клиенте ClickHouse.

1. Измените таблицу (замените `history_json` и 3600 на нужные вам значения):

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Немедленно примените изменения:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Проверьте, что период хранения данных изменился:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
```

```
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Перезапустите сервер Zabbix, чтобы обновить период хранения данных в *Администрирование* > *Обслуживание данных*:

```
systemctl restart zabbix-server
```

Настройка кластеров ClickHouse

В этом разделе описаны шаги настройки кластеров ClickHouse с репликацией и шардингом.

В приведенных ниже шагах используются Docker и ClickHouse `cluster_2S_2R` из документации ClickHouse (запуск ClickHouse 26.4.4.38 с ClickHouse Keeper).

Перед началом рекомендуется разместить перед кластером ClickHouse для Zabbix балансировщик нагрузки. Конфигурация Zabbix server и веб-интерфейса поддерживает только один URL ClickHouse, поэтому она может указывать только на один узел сети. Если этот узел выйдет из строя, балансировщик нагрузки перенаправит Zabbix на исправный узел через тот же URL.

1. Создайте базу данных Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

Attention:
Предложение `ON CLUSTER` затрагивает только те узлы, которые входят в кластер на момент выполнения команды. Если позже вы добавите узлы в кластер, выполните эту команду еще раз.

2. Создайте и настройте пользователя Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Импортируйте схему базы данных с помощью скрипта `history_all.sh` из каталога Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
Период хранения данных (Time-To-Live, или TTL) для ClickHouse по умолчанию составляет 31 день. Чтобы изменить его, используйте параметр `--ttl` при импорте схемы базы данных (например, `--ttl 604800` для 7 дней) или настройте его позже.

Note:
Если вашему кластеру ClickHouse требуются явные пути реплик (например, если в конфигурации сервера ClickHouse не заданы пути по умолчанию для реплицируемых таблиц), запустите скрипты схемы (`history_schema.sh`, `history_uint_schema.sh` и т. д.) по отдельности и задайте для каждой таблицы отдельный путь:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Подробнее о репликации ClickHouse см. в разделах Replicating data и Replicated* table engines в документации ClickHouse.

4. Настройте шардинг для каждого типа таблиц. Переименуйте таблицы и создайте таблицу `Distributed` с исходным именем, указывающую на переименованную таблицу. Именно из этой таблицы `Distributed` Zabbix фактически читает и в нее записывает данные; таблица автоматически распределяет данные по шардам. Например, для таблицы `history`:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Подробнее о шардинге ClickHouse см. в разделе Distributed table engine документации ClickHouse.

5. Настройте Zabbix server.

6. Настройте веб-интерфейс Zabbix.

7. Убедитесь, что данные Zabbix распределены по шардам. Количество строк в `zabbix.history_local` может меняться между повторными проверками, поскольку Zabbix server непрерывно вставляет новые данные. Выполните эту команду на каждом узле:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Проверьте, что репликация синхронизирована (`active_replicas` должно совпадать с `total_replicas`):

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Устранение неполадок

Следующие шаги могут помочь вам устранить проблемы с вашей конфигурацией ClickHouse:

1. Проверьте журналы ClickHouse или сервера Zabbix на наличие ошибок.
2. Чтобы выявить медленные запросы, используйте параметр `log_slow_queries` в параметре конфигурации сервера Zabbix `HistoryProvider`.
3. Убедитесь, что ClickHouse разрешает доступ с сервера Zabbix и веб-интерфейса Zabbix.
4. Выполните запрос к ClickHouse, чтобы проверить, сохраняются ли данные, собранные Zabbix, например:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Примечания по настройке NGINX для Zabbix с учетом особенностей дистрибутива

RHEL

В RHEL 8, 9 и 10 NGINX включён в AppStream — дополнительные репозитории не требуются. Просто включите модуль и установите NGINX:

```
dnf module list nginx
dnf -y install nginx
```

В RHEL 7 NGINX можно установить из репозитория CentOS 7 Extras (если используется CentOS) с помощью команды:

```
yum -y install nginx
```

Если вам нужны самые последние сборки от разработчиков, настройте официальный репозиторий NGINX в `/etc/yum.repos.d/ngin`

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

В SUSE Linux Enterprise Server 15 необходимо настроить `php-fpm` (путь к файлу конфигурации может немного отличаться в зависимости от сервис-пака):

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

Настройка SAML с Microsoft Entra ID

Обзор

В этом разделе приведены рекомендации по настройке единого входа и инициализации пользователей в Zabbix из Microsoft Entra ID (ранее Microsoft Azure Active Directory) с использованием аутентификации SAML 2.0.

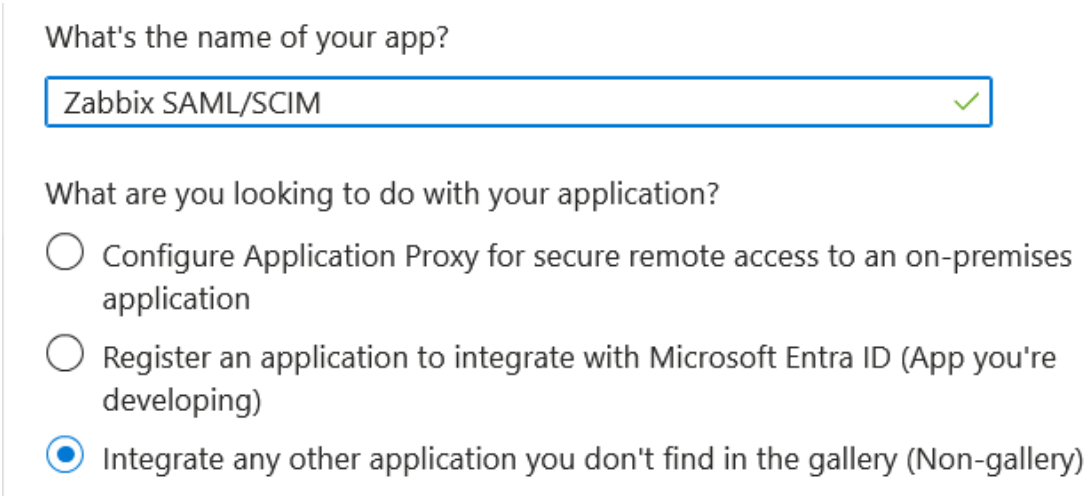
Настройка Microsoft Entra ID

Создание приложения

1. Войдите в центр администрирования Microsoft Entra на Microsoft Entra ID. Для целей тестирования вы можете создать бесплатную пробную учетную запись в Microsoft Entra ID.

2. В центре администрирования Microsoft Entra выберите *Applications* > *Enterprise applications* > *New application* > *Create your own application*.

3. Добавьте имя вашего приложения и выберите параметр *Integrate any other application...*. После этого нажмите *Create*.

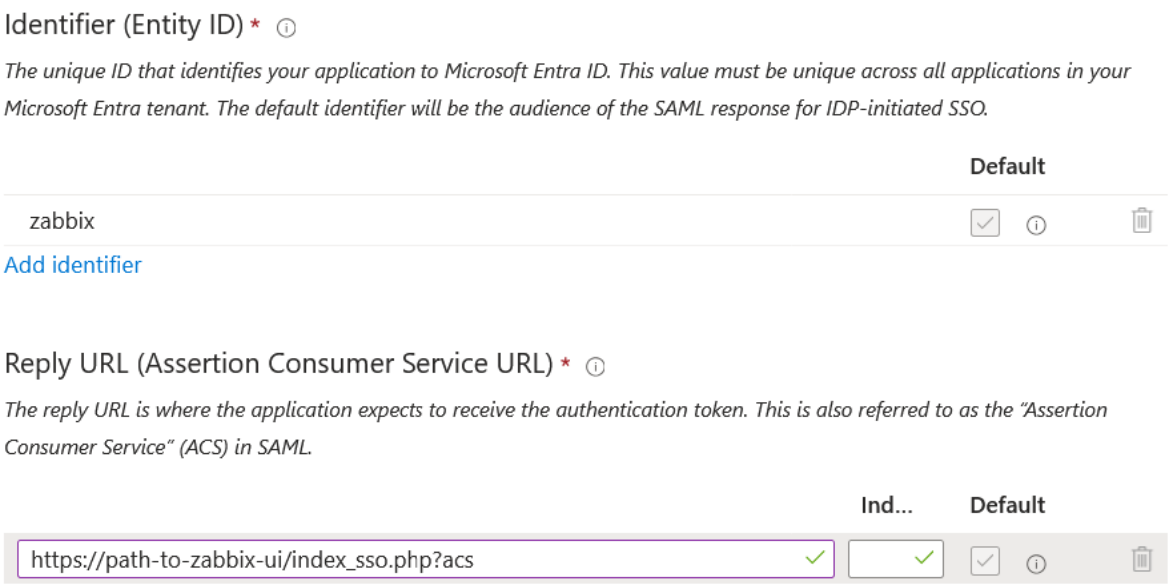


Настройка единого входа

1. На странице приложения перейдите в раздел *Set up single sign on* и нажмите *Get started*. Затем выберите *SAML*.

2. Измените настройки *Basic SAML Configuration*:

- В поле *Identifier (Entity ID)* укажите уникальное имя для идентификации приложения в Microsoft Entra ID, например `zabbix`;
- В поле *Reply URL (Assertion Consumer Service URL)* укажите конечную точку единого входа Zabbix: `https://<path-to-zabbix-`



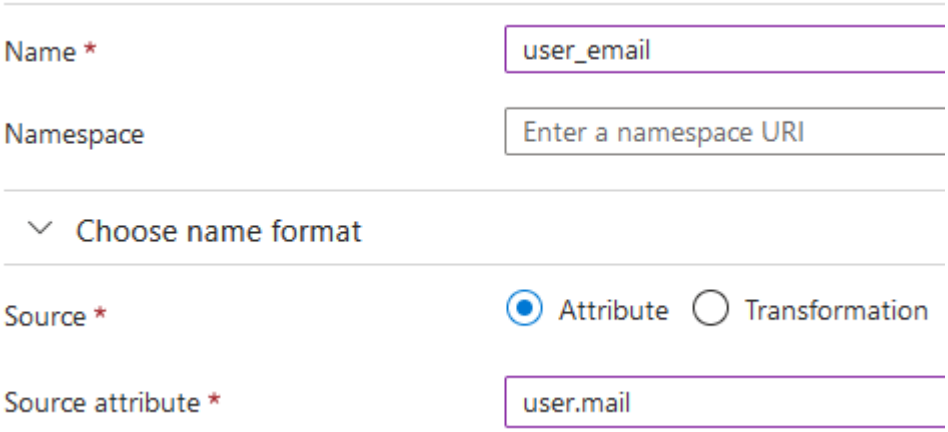
Обратите внимание, что требуется использовать `https`. Чтобы использовать это с Zabbix, добавьте следующую строку в файл конфигурации веб-интерфейса:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

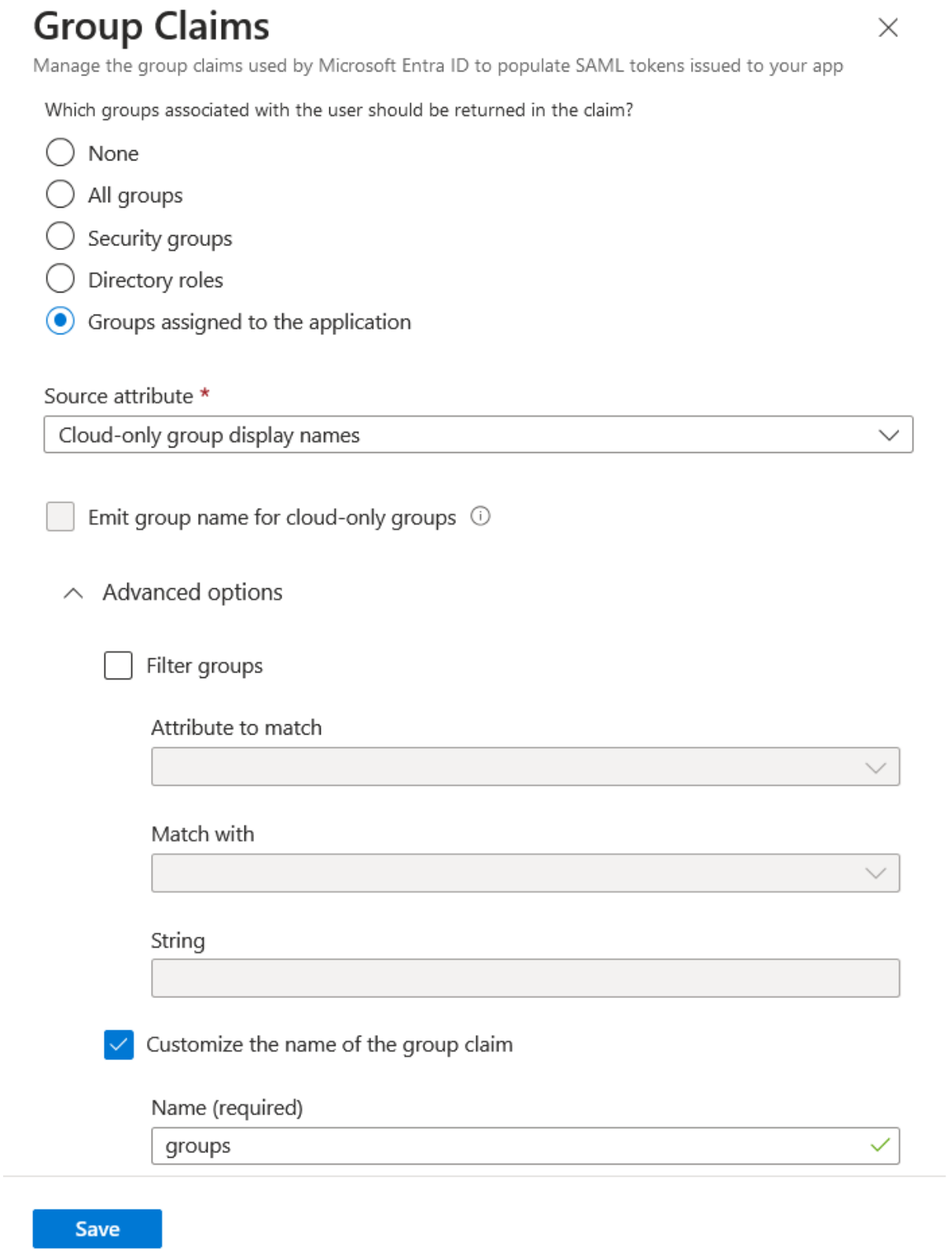
3. Измените настройки *Attributes & Claims*. Необходимо добавить все атрибуты, которые вы хотите передавать в Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Имена атрибутов могут быть произвольными. Можно использовать другие имена атрибутов, однако они должны соответствовать значениям соответствующих полей в настройках SAML Zabbix.

- Нажмите *Add new claim*, чтобы добавить атрибут:



- Нажмите *Add a group claim*, чтобы добавить атрибут для передачи групп в Zabbix:



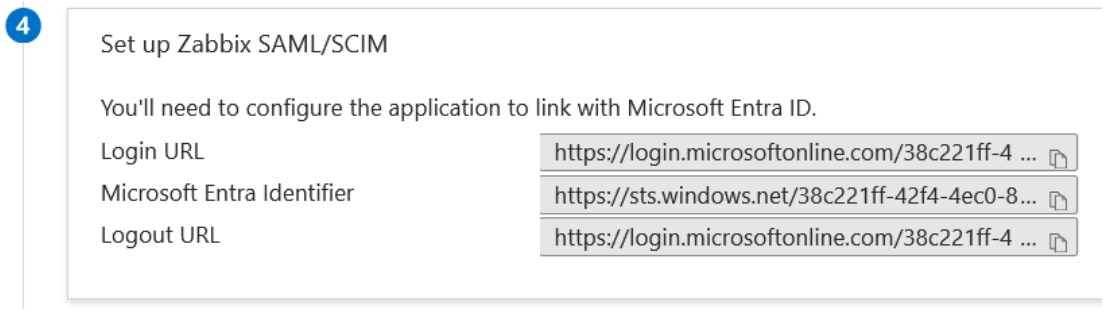
Важно, чтобы в этом утверждении выбранный *Source attribute* передавал в Zabbix имена групп, а не их идентификаторы. В противном случае автоматическое создание пользователей JIT не будет работать должным образом.

4. В разделе *SAML Certificates* скачайте сертификат Base64, предоставленный Entra ID, и поместите его в каталог `conf/certs` установки веб-интерфейса Zabbix (расположение может отличаться в зависимости от ОС и способа установки Zabbix).

Установите для него права доступа 644, выполнив команду:

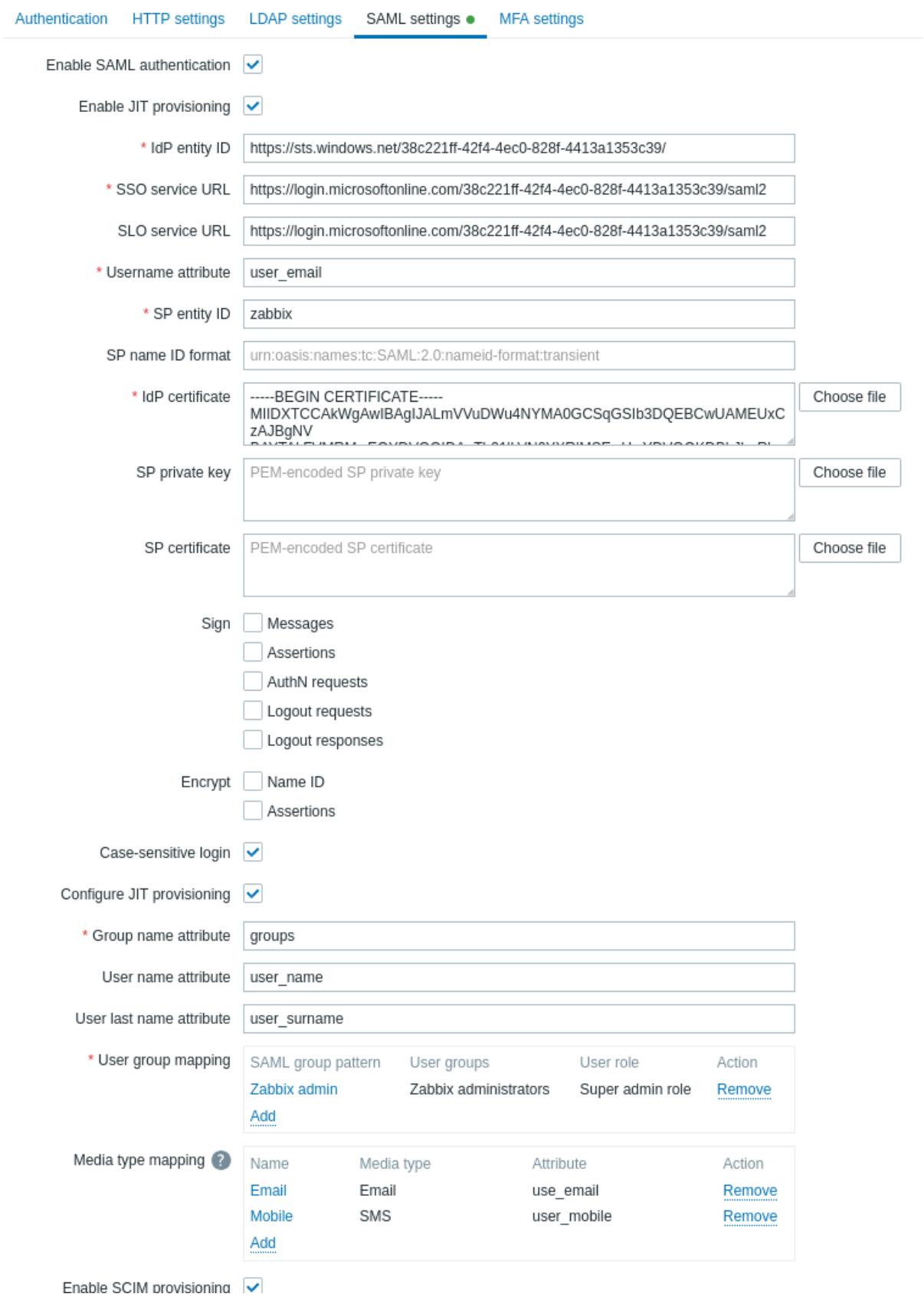
```
chmod 644 entra.cer
```

5. Используйте значения из раздела *Set up <your app name>* в Entra ID для настройки аутентификации Zabbix SAML (см. следующий раздел):



Конфигурация Zabbix

1. В Zabbix перейдите в раздел Настройки SAML и заполните параметры конфигурации на основе конфигурации Entra ID:



Поле Zabbix	Поле настройки в Entra ID	Пример значения
Идентификатор сущности IdP	Идентификатор Microsoft Entra	
URL службы SSO	URL входа	
URL службы SLO	URL выхода	
Идентификатор сущности SP	Идентификатор (Entity ID)	
Атрибут имени пользователя	Пользовательский атрибут (утверждение)	`user_email`
Атрибут имени группы	Пользовательский атрибут (утверждение)	`groups`
Атрибут имени пользователя	Пользовательский атрибут (утверждение)	`user_name`

Поле Zabbix	Поле настройки в Entra ID	Пример значения
Атрибут фамилии пользователя	Пользовательский атрибут (утверждение)	`user_lastname`

2. Добавьте сертификат Base64, предоставленный Entra ID.

Если в файле конфигурации веб-интерфейса (`zabbix.conf.php`) задано `$SSO['CERT_STORAGE'] = 'database'`, во время настройки SAML можно вставить текст сертификата или загрузить файл сертификата в веб-интерфейс — файлы в файловой системе не требуются.

Если в файле конфигурации задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в `ui/conf/certs` или по пути, настроенному в файле конфигурации), а веб-интерфейс не будет сохранять сертификаты в базе данных. Обратите внимание: если `$SSO['CERT_STORAGE']` не задан или закомментирован, используется файловое хранилище, а сертификаты считываются из `ui/conf/certs`.

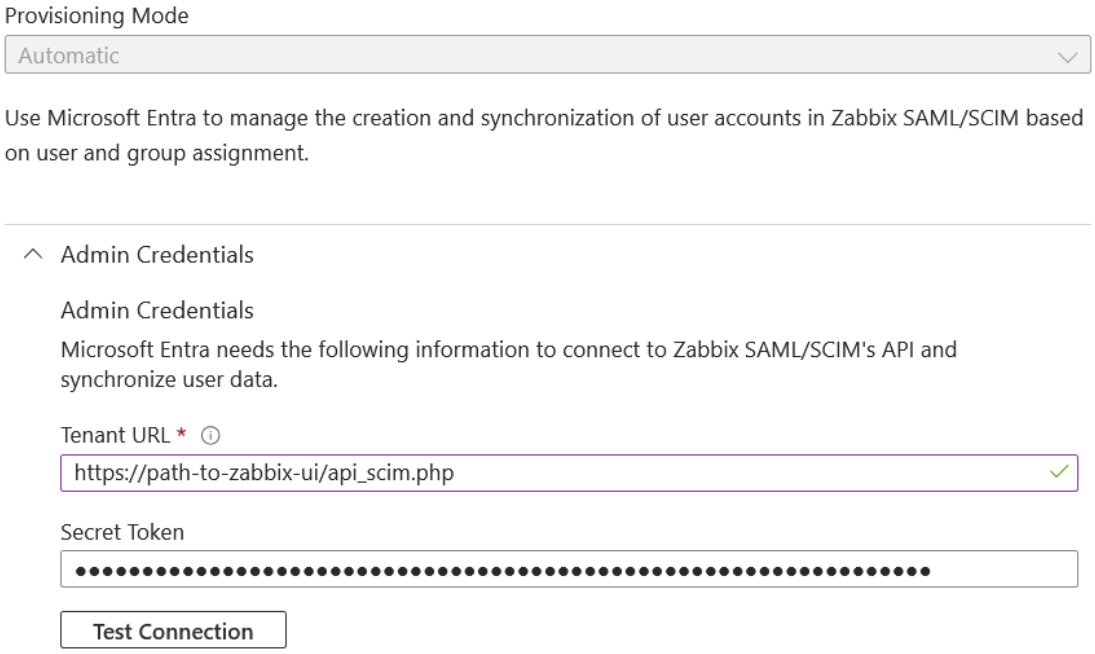
Также необходимо настроить сопоставление групп пользователей. Сопоставление медиа необязательно.

3. Нажмите кнопку *Обновить*, чтобы сохранить эти настройки.

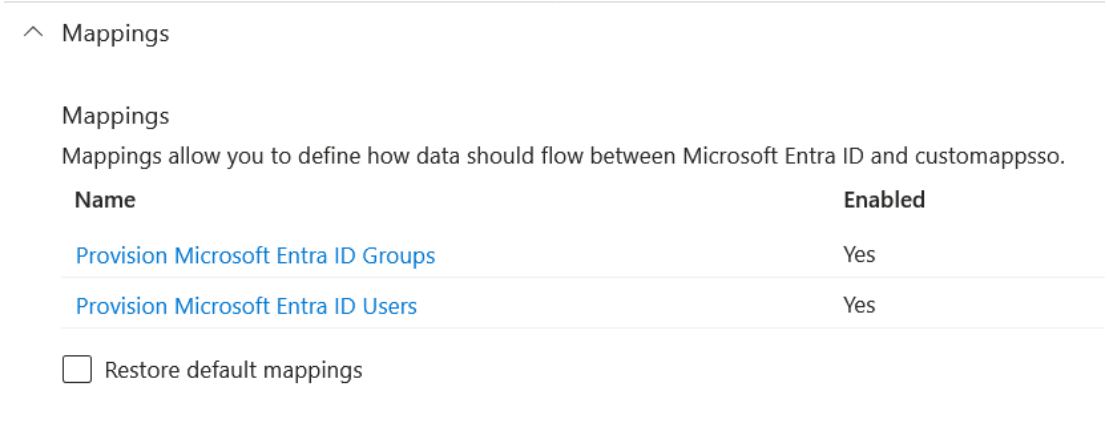
Подготовка пользователей SCIM

1. На странице приложения Entra ID в главном меню откройте страницу Provisioning. Нажмите *Get started* и затем выберите режим Automatic provisioning:

- В *Tenant URL* укажите следующее значение: `https://<path-to-zabbix-ui>/api_scim.php`
- В *Secret token* введите токен API Zabbix с правами Super admin.
- Нажмите *Test connection*, чтобы проверить, установлено ли соединение.

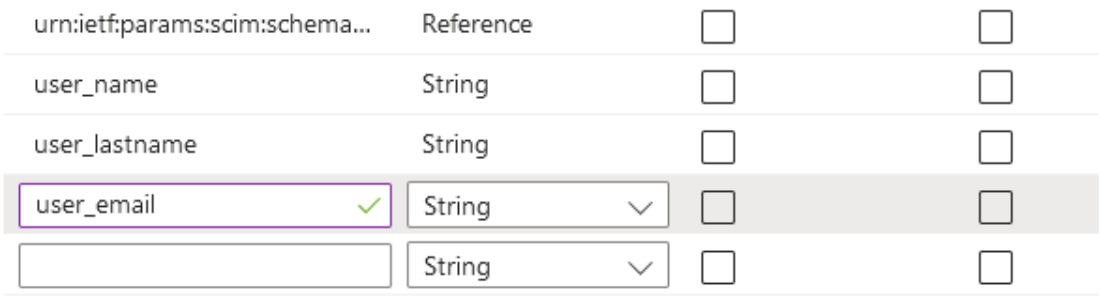


2. Теперь можно добавить все атрибуты, которые будут передаваться через SCIM в Zabbix. Для этого нажмите *Mappings*, а затем *Provision Microsoft Entra ID Users*.



В нижней части списка Attribute Mapping включите *Show advanced options*, а затем нажмите *Edit attribute list for custom-appsso*.

В нижней части списка атрибутов добавьте собственные атрибуты с типом 'String':



Сохраните список.

3. Теперь можно добавить сопоставления для добавленных атрибутов. В нижней части списка Attribute Mapping нажмите *Add New Mapping* и создайте сопоставления, как показано ниже:



После добавления всех сопоставлений сохраните список сопоставлений.



4. В качестве предварительного условия для подготовки пользователей в Zabbix у вас должны быть настроены пользователи и группы в Entra ID.

Для этого перейдите в *Microsoft Entra admin center*, а затем добавьте пользователей/группы на соответствующих страницах Users и Groups.

5. После создания пользователей и групп в Entra ID перейдите в меню *Users and groups* вашего приложения и добавьте их в приложение.

6. Перейдите в меню *Provisioning* вашего приложения и нажмите *Start provisioning*, чтобы выполнить подготовку пользователей в Zabbix.

Обратите внимание, что запрос Users PATCH в Entra ID не поддерживает изменения в media.

Подпись запросов аутентификации

Entra ID можно настроить для проверки подписи подписанных запросов аутентификации.

Чтобы это работало, создайте открытый и закрытый ключи:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Назначьте разрешения:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Обновите конфигурацию веб-интерфейса Zabbix, добавив:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Устранение неполадок

При использовании браузера Microsoft Edge могут возникать проблемы с аутентификацией, если пользователь, пытающийся войти в Zabbix через SAML, уже вошел в профиль Microsoft Edge. Признаком такой проблемы может быть возможность войти в Zabbix через Microsoft Edge в приватном режиме.

Чтобы избежать проблем с аутентификацией в этом случае, может потребоваться установить для `requestedAuthnContext` значение `false` в файле конфигурации веб-интерфейса Zabbix.

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

Настройка SAML с Okta

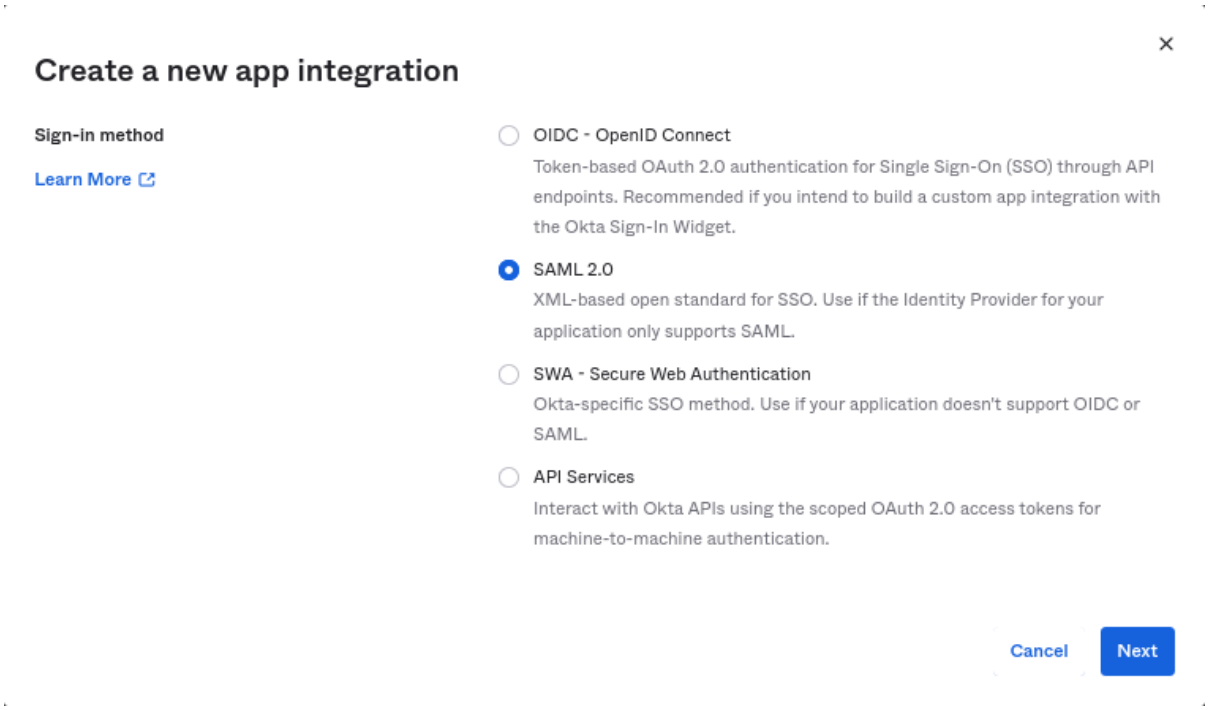
В этом разделе приведены рекомендации по настройке Okta для включения аутентификации SAML 2.0 и подготовки пользователей для Zabbix.

Настройка Okta

1. Перейдите на https://developer.okta.com/signup/ и зарегистрируйтесь/войдите в свою учетную запись.

2. В веб-интерфейсе Okta перейдите в *Applications* > *Applications*.

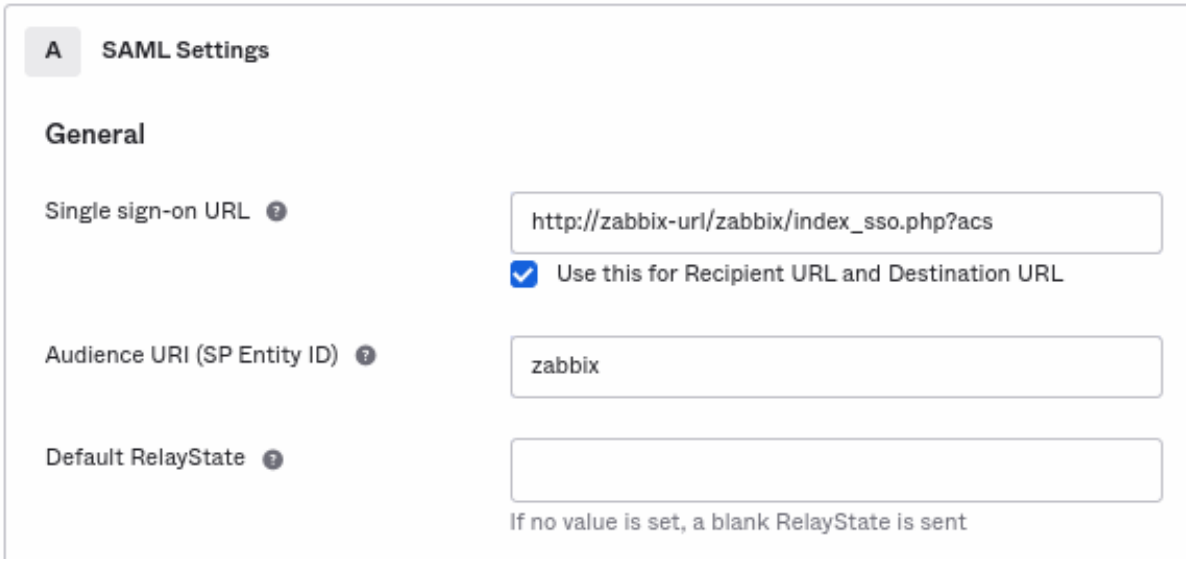
3. Нажмите *Create App Integration*.



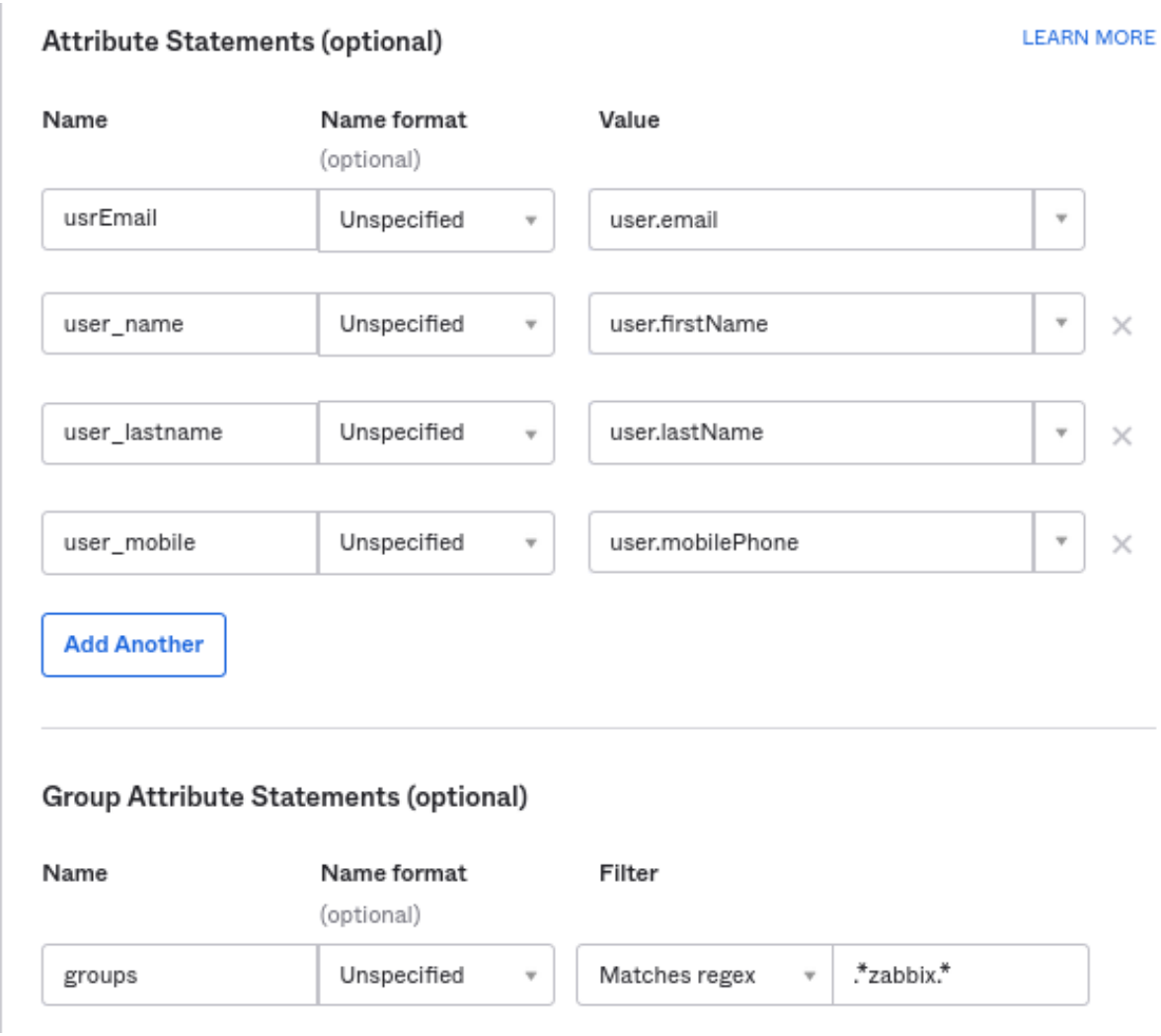
Выберите "SAML 2.0" в качестве метода входа и нажмите *Next*.

4. В общих настройках укажите имя приложения и нажмите *Next*.

5. В настройках SAML введите значения, указанные ниже, затем нажмите *Next*.



- В разделе **General** добавьте:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Обратите внимание на использование "http", а не "https", чтобы параметр `acs` не был обрезан в запросе. Флажок *Use this for Recipient URL and Destination URL* также должен быть установлен.
  - *Audience URI (SP Entity ID)*: `zabbix<br>` Обратите внимание, что это значение будет использоваться в SAML assertion как уникальный идентификатор поставщика услуг (если оно не совпадает, операция будет отклонена). В этом поле можно указать URL или любую строку данных.
  - *Default RelayState*:
 Оставьте это поле пустым; если требуется пользовательское перенаправление, его можно добавить в Zabbix в настройках *Users* > *Users*.
  - Заполните остальные поля по своему усмотрению.
- В **Attribute Statements/Group Attribute Statements** добавьте:



Эти атрибуты вставляются в SAML assertion, которыми обмениваются с Zabbix.

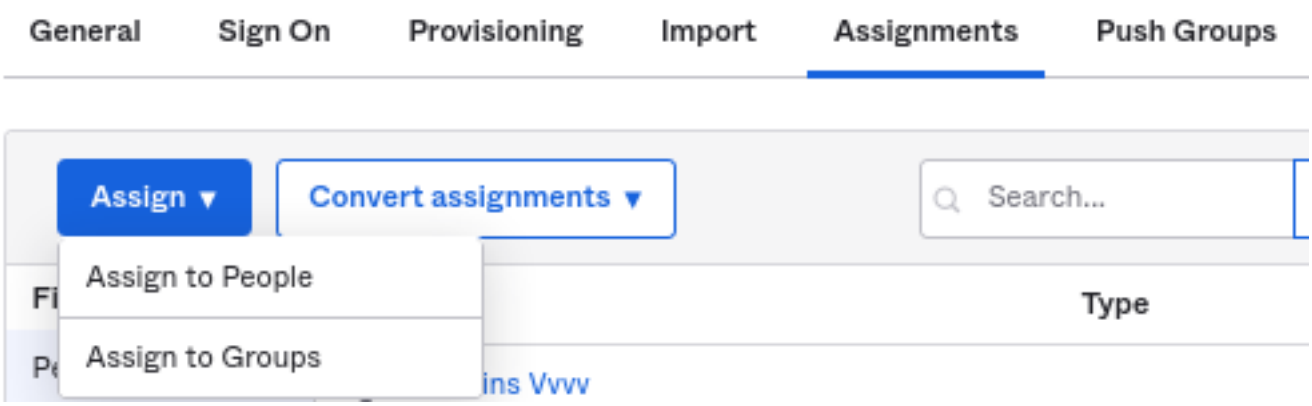
Имена атрибутов, используемые здесь, приведены в качестве произвольных примеров. Вы можете использовать другие имена атрибутов, однако они должны совпадать со значением соответствующего поля в настройках Zabbix SAML.

Если вы хотите настроить вход в Zabbix по SAML *без* JIT user provisioning, тогда требуется только атрибут email.

Note:
Если планируется использовать зашифрованное соединение, сгенерируйте закрытый и открытый сертификаты шифрования, затем загрузите открытый сертификат в Okta. Форма загрузки сертификата появляется, когда для *Assertion Encryption* установлено значение "Encrypted" (нажмите *Show Advanced Settings*, чтобы найти этот параметр).

6. На следующей вкладке выберите "I'm a software vendor. I'd like to integrate my app with Okta" и нажмите *Finish*.

7. Перейдите на вкладку "Assignments" только что созданного приложения и нажмите кнопку *Assign*, затем выберите *Assign to People* в раскрывающемся списке.



8. Во всплывающем окне назначьте приложение пользователям, которые будут использовать SAML 2.0 для аутентификации в Zabbix, затем нажмите *Save and go back*.

9. Скачайте сертификат IdP. Установите для него права 644, выполнив:

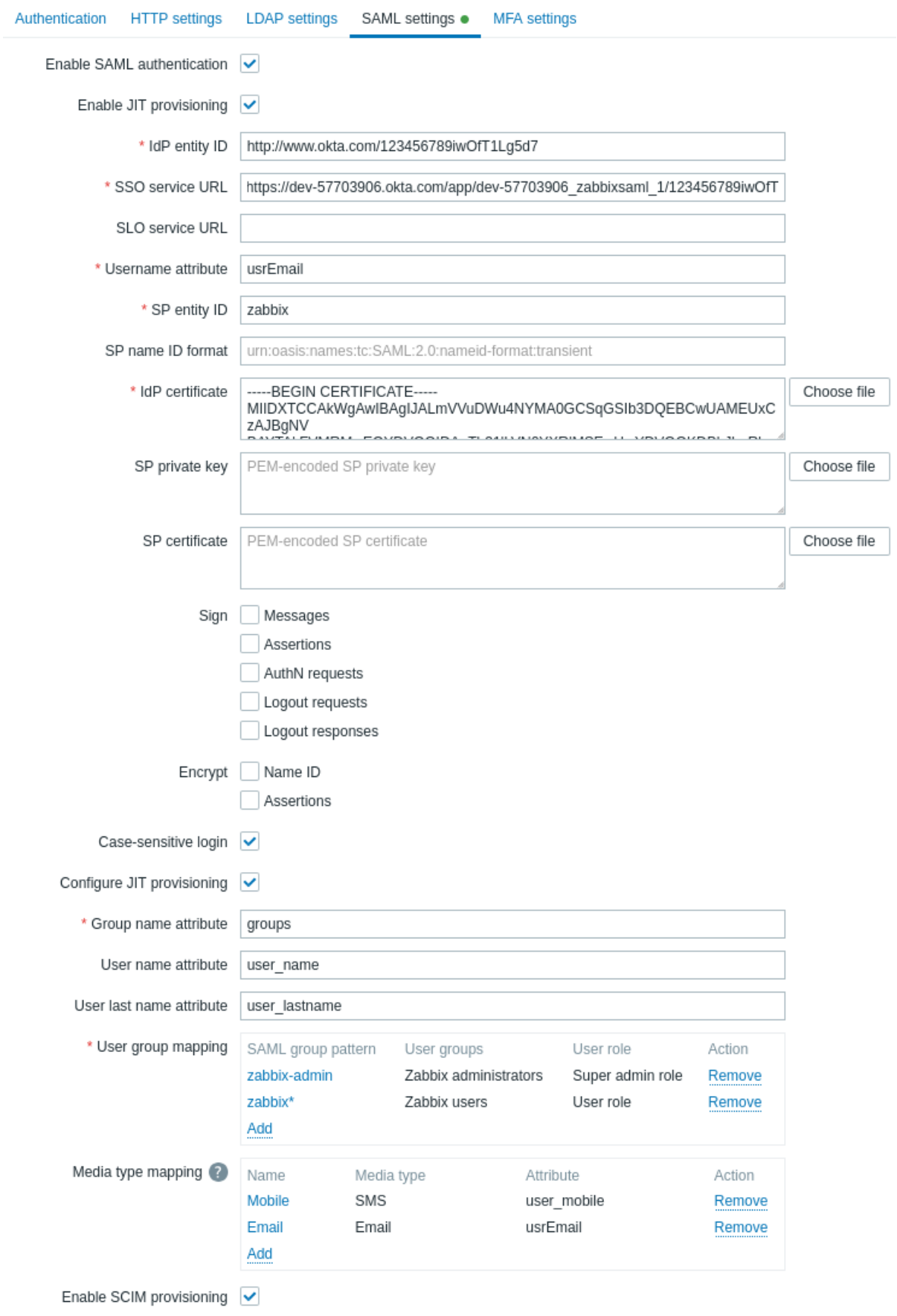
```
chmod 644 idp.crt
```

10. Перейдите на вкладку "Sign On" и нажмите кнопку *View Setup Instructions*.

Инструкции по настройке откроются в новой вкладке; оставьте эту вкладку открытой во время настройки Zabbix.

Конфигурация Zabbix

1. В Zabbix перейдите в раздел Настройки SAML и заполните параметры конфигурации на основе инструкций по настройке в Okta:



Поле Zabbix	Поле настройки в Okta	Пример значения
Идентификатор сущности IdP	Identity Provider Issuer	
URL службы SSO	Identity Provider Single Sign-On URL	
Атрибут имени пользователя	Attribute name	`usrEmail`
Идентификатор сущности SP	Audience URI	`zabbix`
Атрибут имени группы	Attribute name	`groups`

Поле Zabbix	Поле настройки в Okta	Пример значения
Атрибут имени пользователя	Attribute name	`user_name`
Атрибут фамилии пользователя	Attribute name	`user_lastname`

Также необходимо настроить сопоставление групп пользователей и способов оповещения.

2. Добавьте сертификат в формате Base64, предоставленный в инструкциях по настройке SAML в Okta.

Если в файле конфигурации веб-интерфейса (`zabbix.conf.php`) задано `$SSO['CERT_STORAGE'] = 'database'`, текст сертификата можно вставить или загрузить файл сертификата в веб-интерфейсе во время настройки SAML — файлы в файловой системе не требуются.

Если в файле конфигурации задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в `ui/conf/certs` или по пути, настроенному в файле конфигурации), а веб-интерфейс не будет сохранять сертификаты в базе данных. Обратите внимание: если `$SSO['CERT_STORAGE']` не задан или закомментирован, используется файловое хранилище, а сертификаты считываются из `ui/conf/certs`.

3. Если в Okta для параметра *Assertion Encryption* установлено значение «Encrypted», в Zabbix также необходимо установить флажок «Assertions» для параметра *Encrypt*.

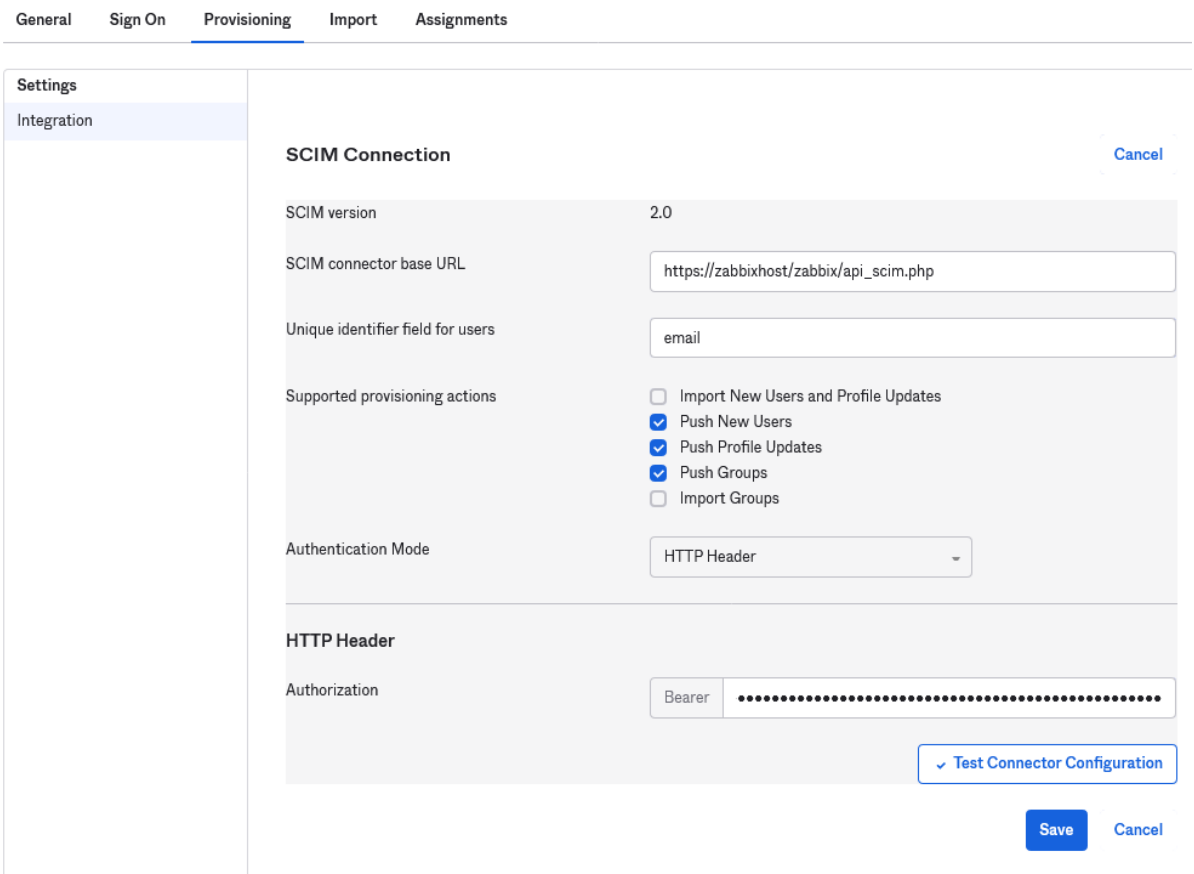
4. Нажмите кнопку *Update*, чтобы сохранить эти настройки.

Настройка SCIM

1. Чтобы включить настройку SCIM, перейдите в раздел *General* > *App Settings* приложения в Okta.

Установите флажок *Enable SCIM provisioning*. В результате появится новая вкладка *Provisioning*.

2. Перейдите на вкладку *Provisioning*, чтобы настроить SCIM-подключение: - В поле *SCIM connector base URL* укажите путь к веб-интерфейсу Zabbix и добавьте к нему `api_scim.php`, например:
 `https://<your-zabbix-url>/zabbix/api_scim.php` - *Unique identifier field for users*: `email` - *Authentication mode*: `HTTP header` - В поле *Authorization* введите действительный API-токен с правами супер-администратора.



Attention:
Если у вас возникают проблемы с аутентификацией, см. раздел Перенаправление заголовка авторизации.

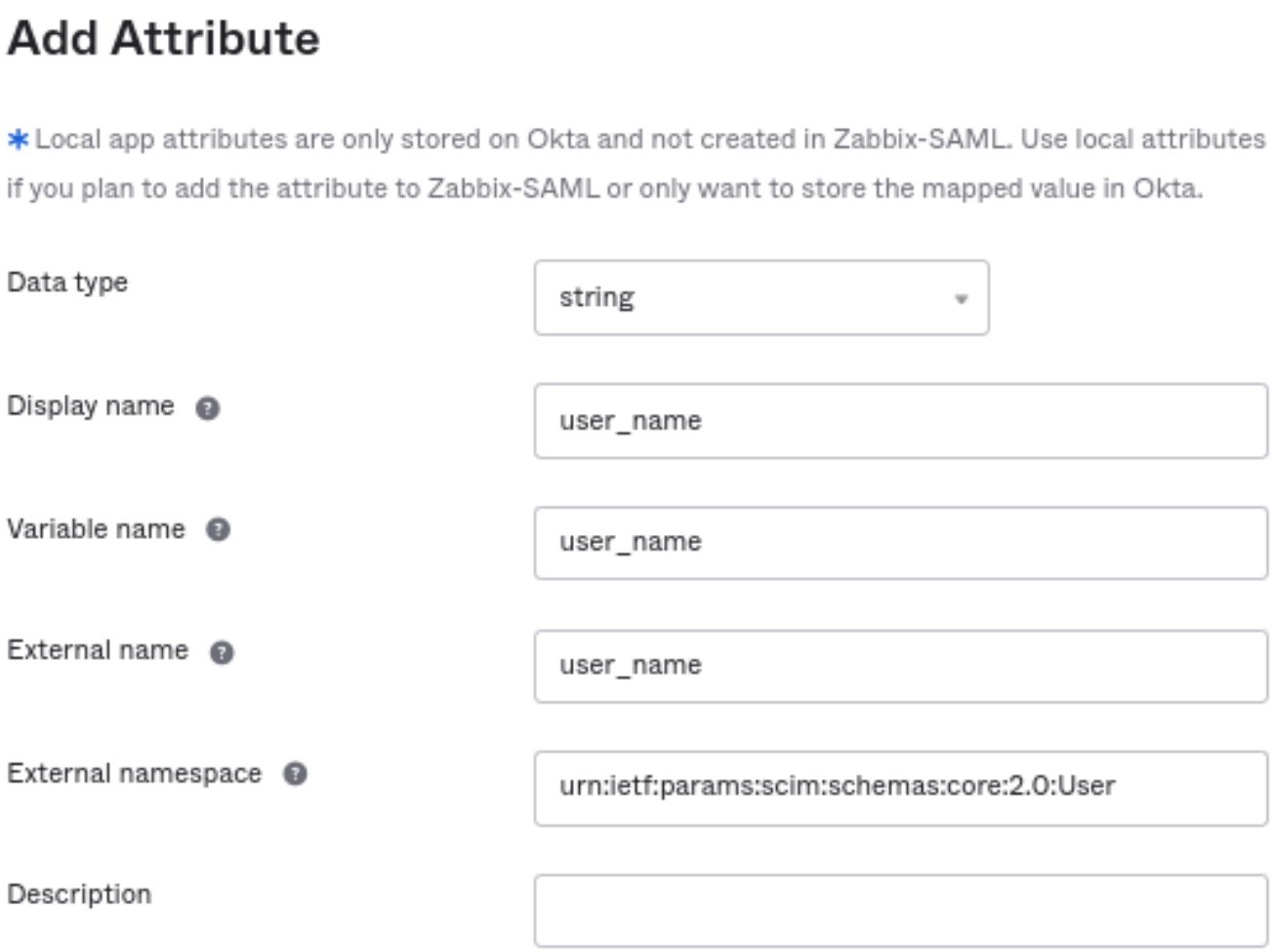
3. Нажмите *Test Connector Configuration*, чтобы проверить подключение. Если все настроено правильно, появится сообщение об успешном выполнении.

4. В разделе *Provisioning* > *To App* установите следующие флажки: - Create Users - Update User Attributes - Deactivate Users

Это гарантирует отправку запросов этих типов в Zabbix.

5. Убедитесь, что все атрибуты, определенные в SAML, определены в SCIM. Редактор профиля приложения можно открыть в разделе *Provisioning* > *To App*, нажав *Go to Profile Editor*.

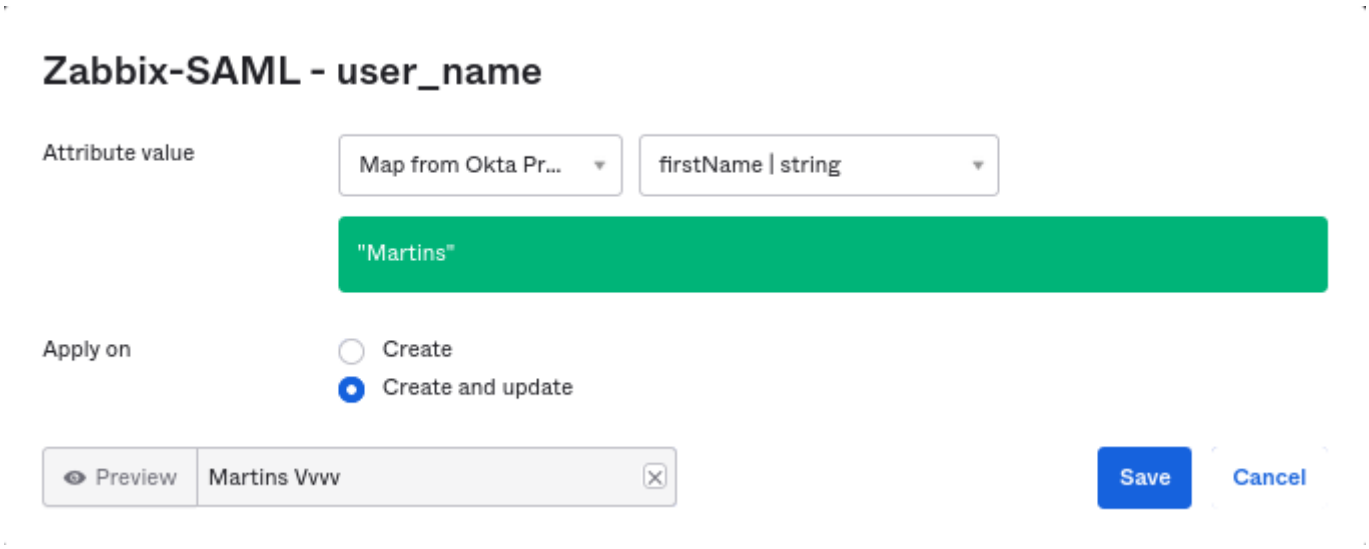
Нажмите *Add Attribute*. Заполните поля *Display name*, *Variable name* и *External name*, указав имя атрибута SAML, например `user_name`.



External namespace должен совпадать со схемой пользователя: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Перейдите в разделе *Provisioning* > *To App* > *Attribute Mappings* приложения. В нижней части страницы нажмите *Show Unmapped Attributes*. Появятся недавно добавленные атрибуты.

7. Сопоставьте каждый добавленный атрибут.



8. Добавьте пользователей на вкладке *Assignments*. Предварительно пользователей необходимо добавить в разделе *Directory* > *People*. Все эти назначения будут отправлены в Zabbix в виде запросов.

9. Добавьте группы на вкладке *Push Groups*. Шаблон сопоставления групп пользователей в настройках SAML Zabbix должен соответствовать группе, указанной здесь. Если соответствия нет, пользователь не может быть создан в Zabbix.

Информация об участниках групп отправляется каждый раз при внесении изменений.

Настройка SAML с OneLogin

Обзор

В этом разделе приведены рекомендации по настройке единого входа и инициализации пользователей в Zabbix из OneLogin с использованием аутентификации SAML 2.0.

Настройка OneLogin

Создание приложения

1. Войдите в свою учетную запись в OneLogin. Для тестирования вы можете создать бесплатную учетную запись разработчика в OneLogin.

2. В веб-интерфейсе OneLogin перейдите в *Applications* > *Applications*.

3. Нажмите "Add App" и найдите подходящее приложение. Рекомендации на этой странице основаны на примере приложения *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

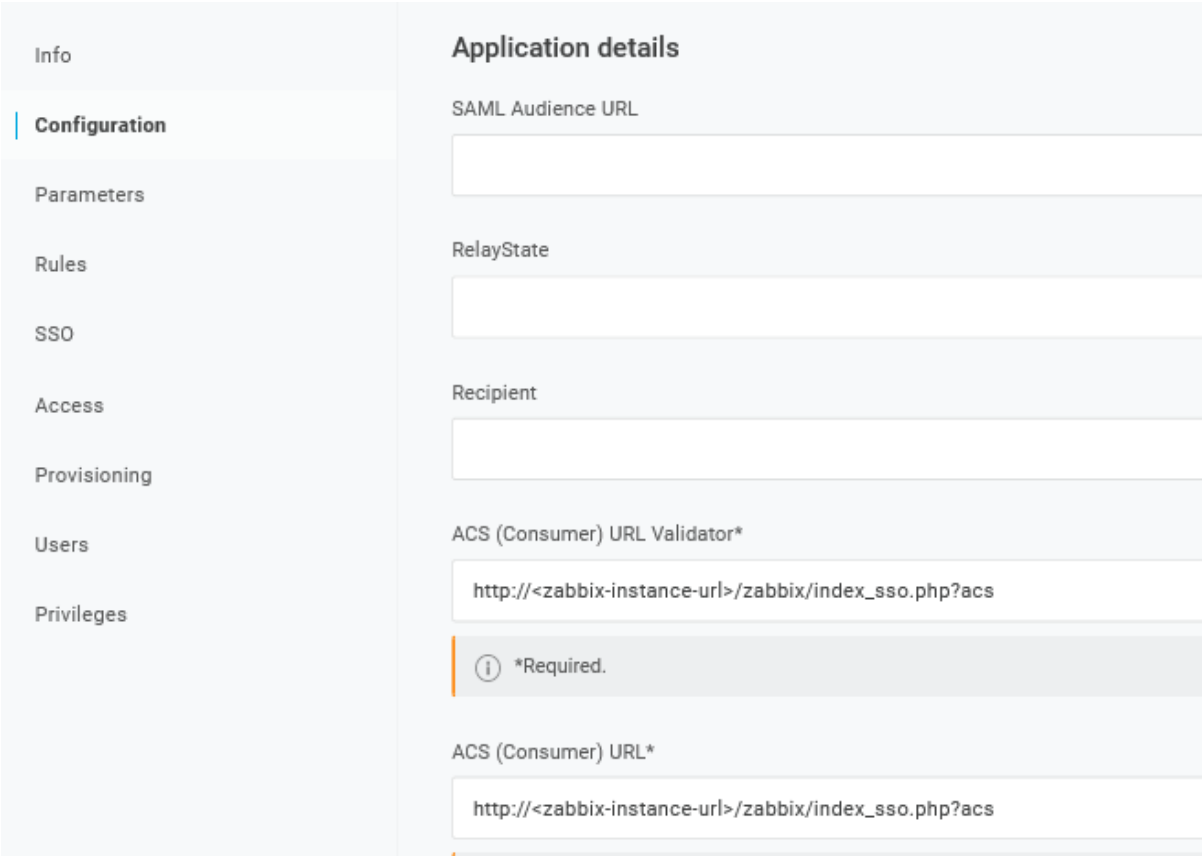
4. Для начала вы можете настроить отображаемое имя вашего приложения. Также вы можете добавить значок и сведения о приложении. После этого нажмите *Save*.

Настройка подготовки SSO/SCIM

1. В разделе *Configuration* > *Application details* укажите конечную точку единого входа Zabbix `http://<zabbix-instance-url>/zab` в следующих полях:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Обратите внимание на использование `http`, а не `https`, чтобы параметр `acs` не был удалён из запроса.



Также можно использовать `https`. Чтобы использовать этот вариант с Zabbix, добавьте следующую строку в файл конфигурации веб-интерфейса (`zabbix.conf.php`):

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Оставьте остальные параметры со значениями по умолчанию.

2. В разделе *Configuration* > *API connection* укажите следующие значения:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: должен содержать все пользовательские атрибуты, которые необходимо передавать в Zabbix через SCIM, например `user_name`, `user_lastname`, `user_email` и `user_mobile`:

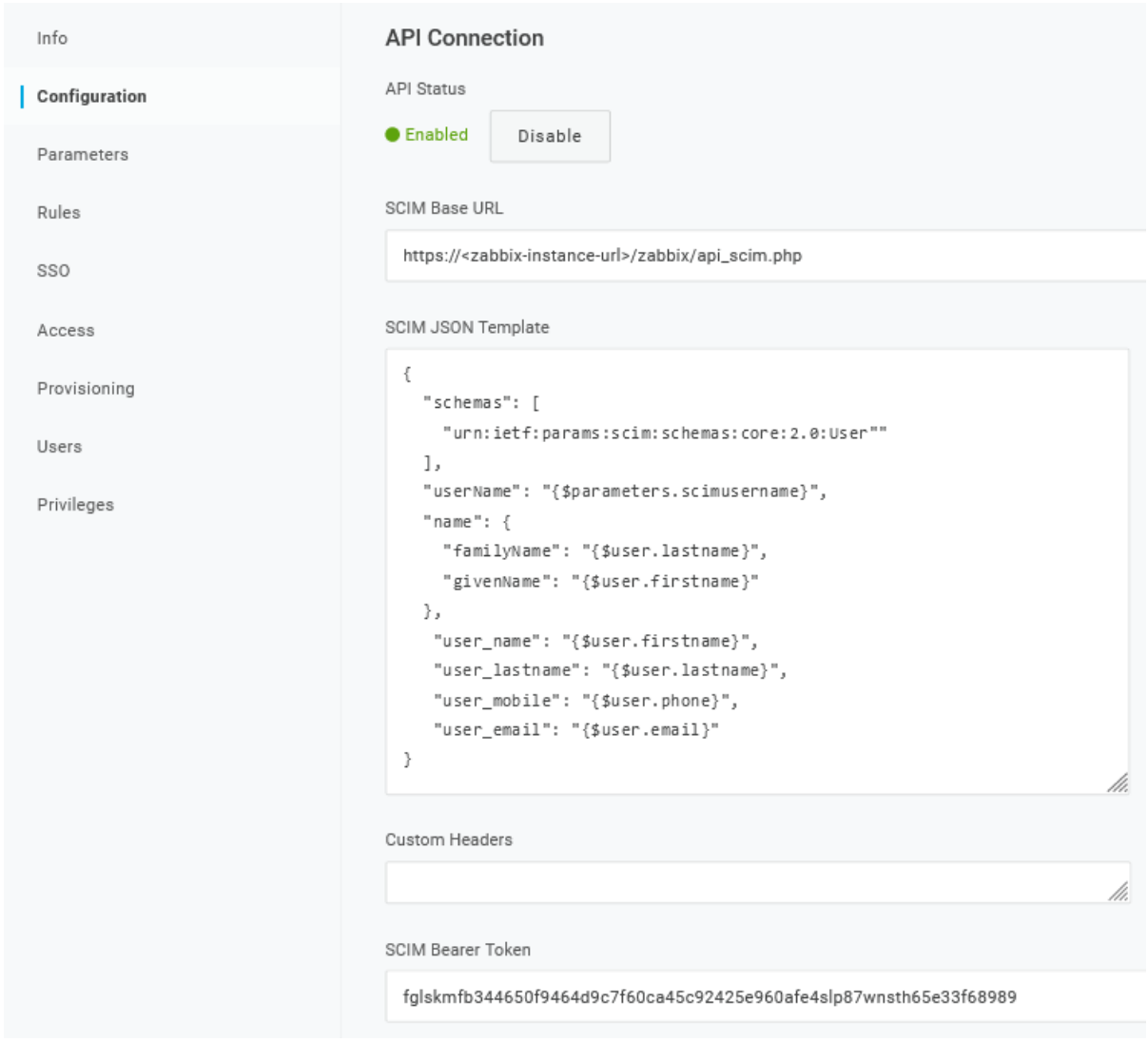
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Имена атрибутов могут быть произвольными. Можно использовать другие имена атрибутов, однако они должны соответствовать значениям соответствующих полей в настройках SAML Zabbix.

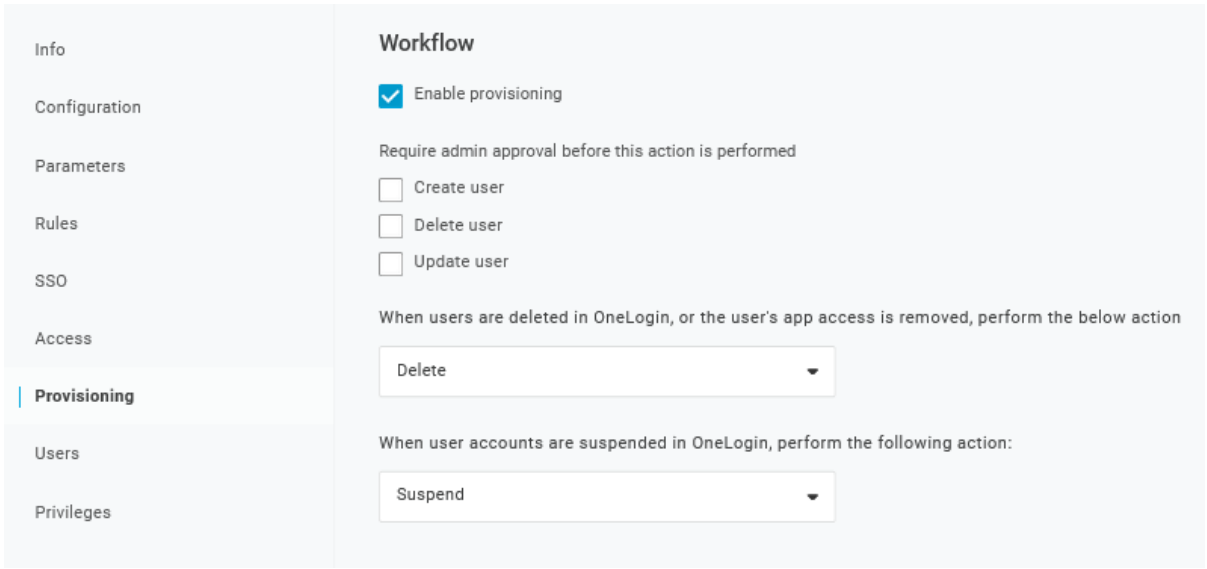
Обратите внимание: для работы подготовки пользователей OneLogin должен получить в ответе атрибут `name` с атрибутами `givenName` и `familyName`, даже если поставщик услуг их не запрашивал. Поэтому необходимо указать их в схеме в разделе конфигурации приложения.

- *SCIM Bearer Token*: введите API-токен Zabbix с правами супер-администратора.

Нажмите *Enable*, чтобы активировать подключение.

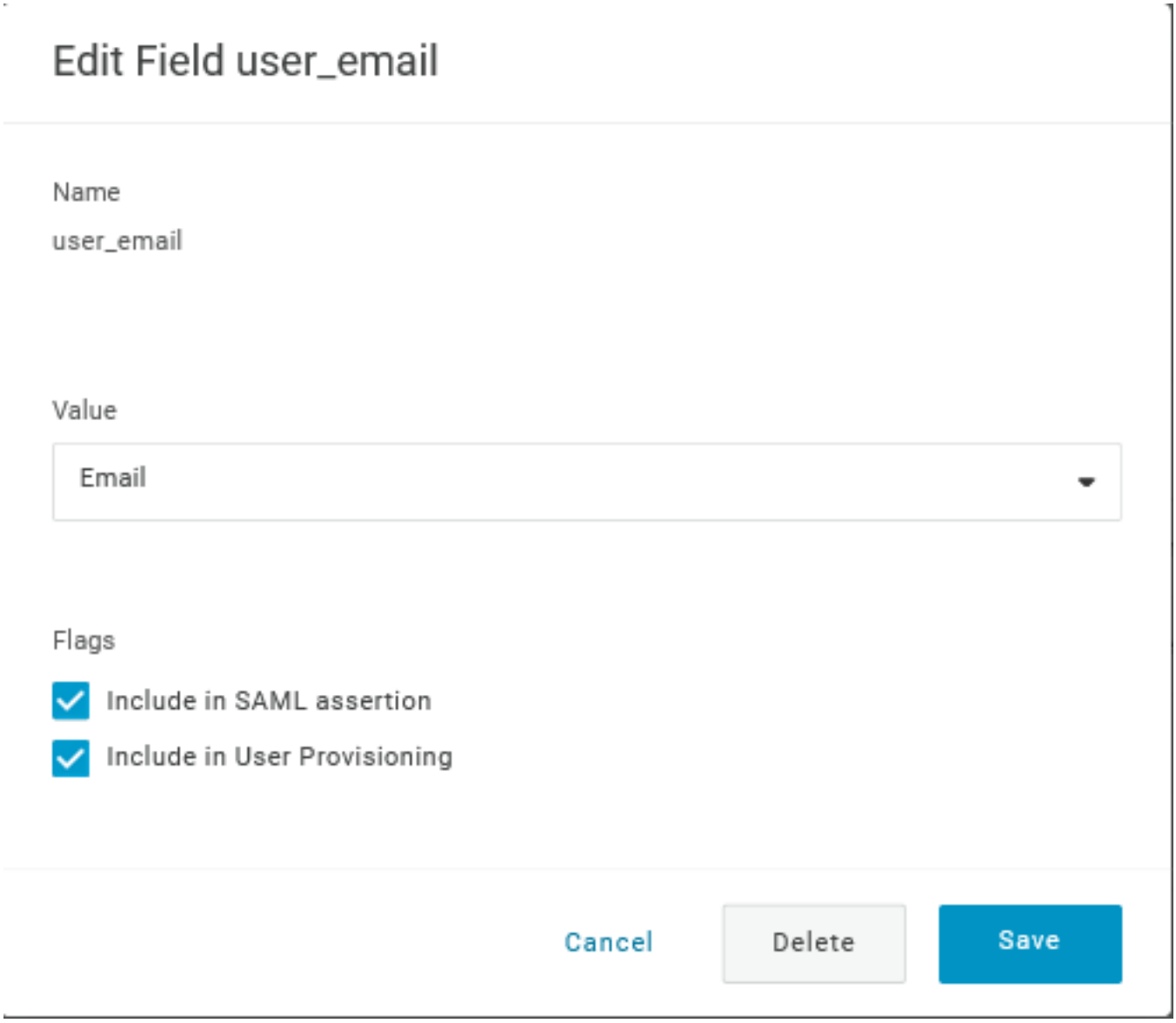


3. На странице *Provisioning* включите параметр Provisioning:



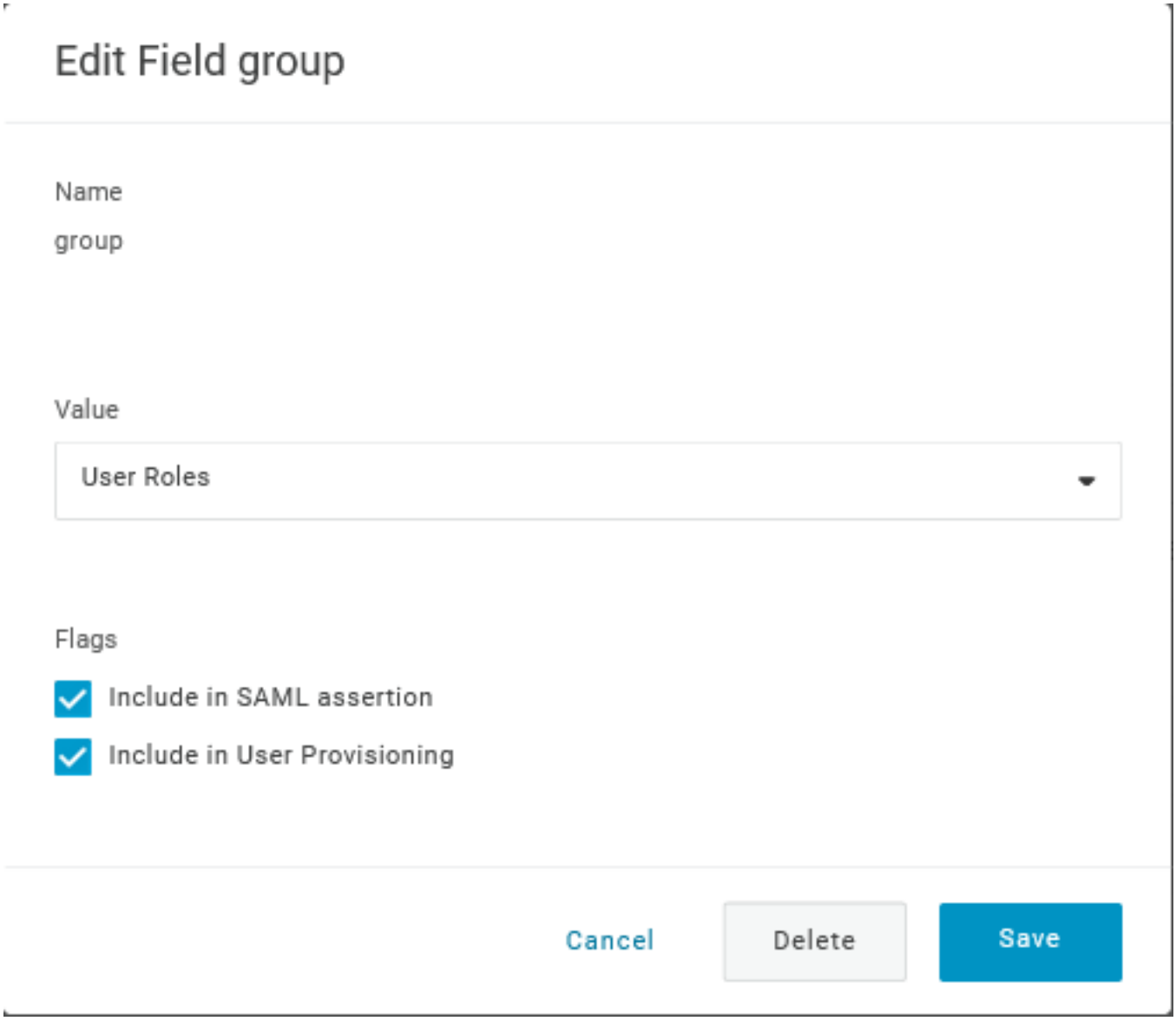
4. На странице *Parameters* содержится список параметров по умолчанию:

- Убедитесь, что `scimusername` соответствует значению логина пользователя в OneLogin (например, адресу электронной почты);
- Для параметра `Groups` включите параметр *Include in User Provisioning*;
- Нажмите «+», чтобы создать пользовательские параметры, необходимые для утверждений SAML и подготовки пользователей, например `user_name`, `user_lastname`, `user_email` и `user_mobile`:

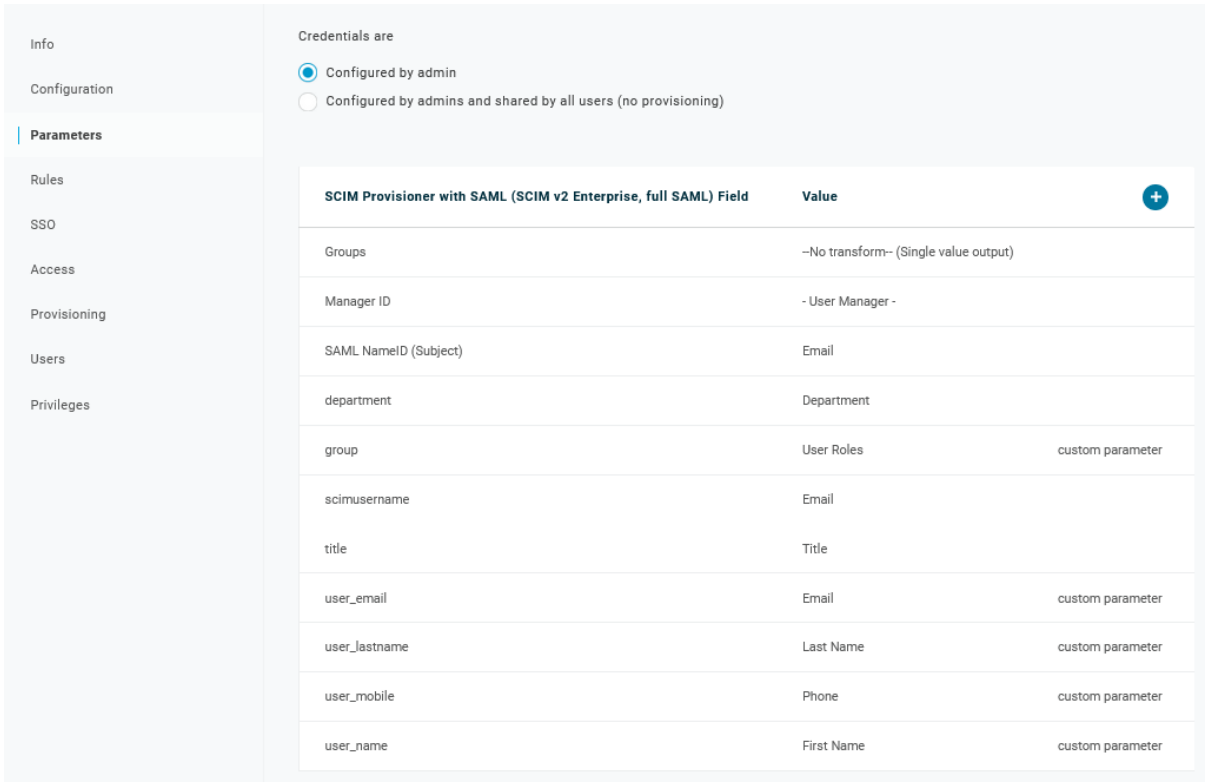


При добавлении параметра обязательно включите оба параметра: *Include in SAML assertion* и *Include in User Provisioning*.

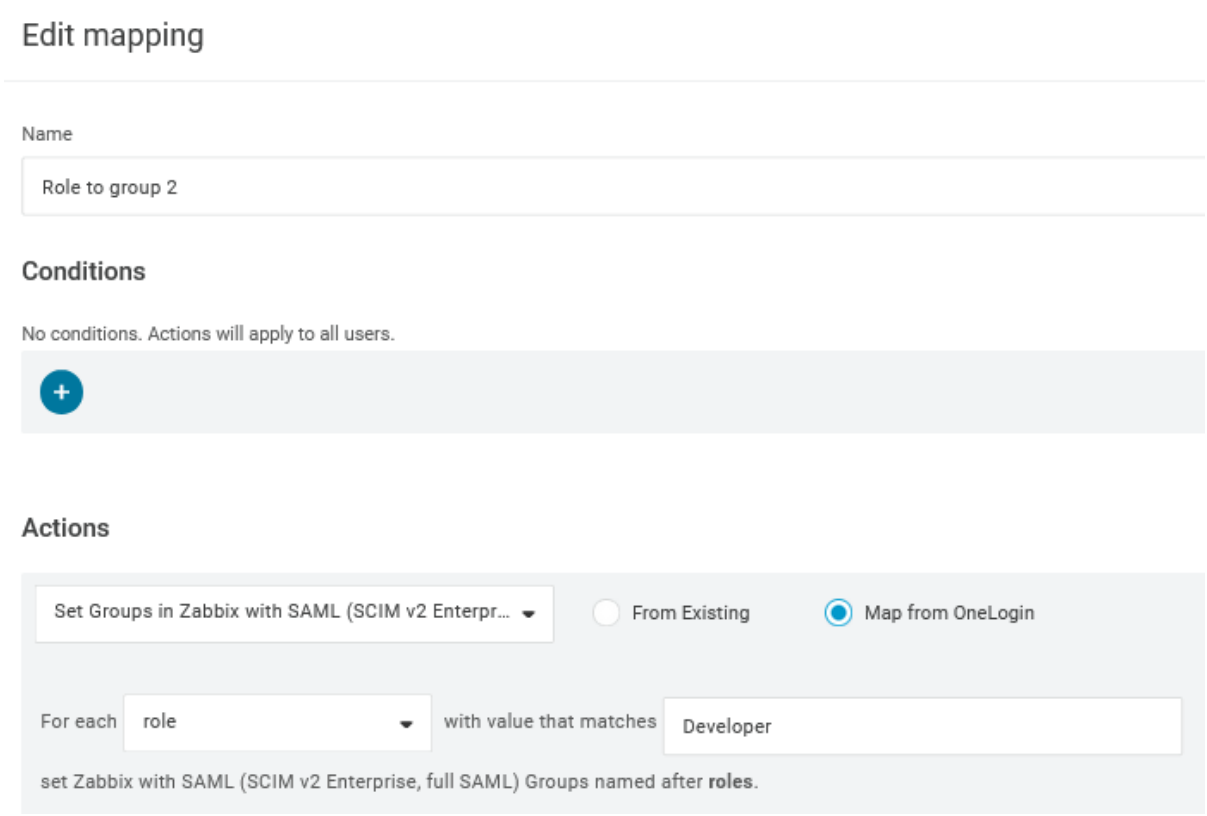
- Добавьте параметр `group`, соответствующий ролям пользователей в OneLogin. Роли пользователей будут передаваться в виде строки, разделённой точкой с запятой `;`. Роли пользователей OneLogin будут использоваться для создания групп пользователей в Zabbix:



Проверьте список параметров:



5. На странице *Rules* создайте сопоставления ролей пользователей с параметром Groups по умолчанию.



Для передачи определённых ролей в качестве групп можно использовать регулярное выражение. Имена ролей не должны содержать `;`, поскольку OneLogin использует этот символ как разделитель при отправке атрибута с несколькими ролями.

6. Скачайте сертификат IdP. Установите для него права 644, выполнив команду:

```
chmod 644 idp.crt
```

Конфигурация Zabbix

1. В Zabbix перейдите в раздел Настройки SAML и заполните параметры конфигурации на основе конфигурации OneLogin:



Поле Zabbix	Поле настройки в OneLogin	Пример значения
Идентификатор сущности IdP	URL издателя (см. вкладку SSO приложения в OneLogin)	
URL службы SSO	Конечная точка SAML 2.0 (HTTP) (см. вкладку SSO приложения в OneLogin)	

Поле Zabbix	Поле настройки в OneLogin	Пример значения
URL службы SLO	Конечная точка SLO (HTTP) (см. вкладку SSO приложения в OneLogin)	
Атрибут имени пользователя	Пользовательский параметр	`user_email`
Атрибут имени группы	Пользовательский параметр	`group`
Атрибут имени пользователя	Пользовательский параметр	`user_name`
Атрибут фамилии пользователя	Пользовательский параметр	`user_lastname`

Также необходимо настроить сопоставление групп пользователей. Сопоставление медиа необязательно. Нажмите *Обновить*, чтобы сохранить эти настройки.

2. Добавьте сертификат Base64, предоставленный OneLogin.

Если в файле конфигурации веб-интерфейса (`zabbix.conf.php`) задано `$SSO['CERT_STORAGE'] = 'database'`, текст сертификата можно вставить или загрузить файл сертификата в веб-интерфейсе во время настройки SAML — файлы в файловой системе не требуются.

Если в файле конфигурации задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в `ui/conf/certs` или по пути, указанному в файле конфигурации), а веб-интерфейс не будет сохранять сертификаты в базе данных. Обратите внимание: если `$SSO['CERT_STORAGE']` не задан или закомментирован, используется файловое хранилище, а сертификаты считываются из `ui/conf/certs`.

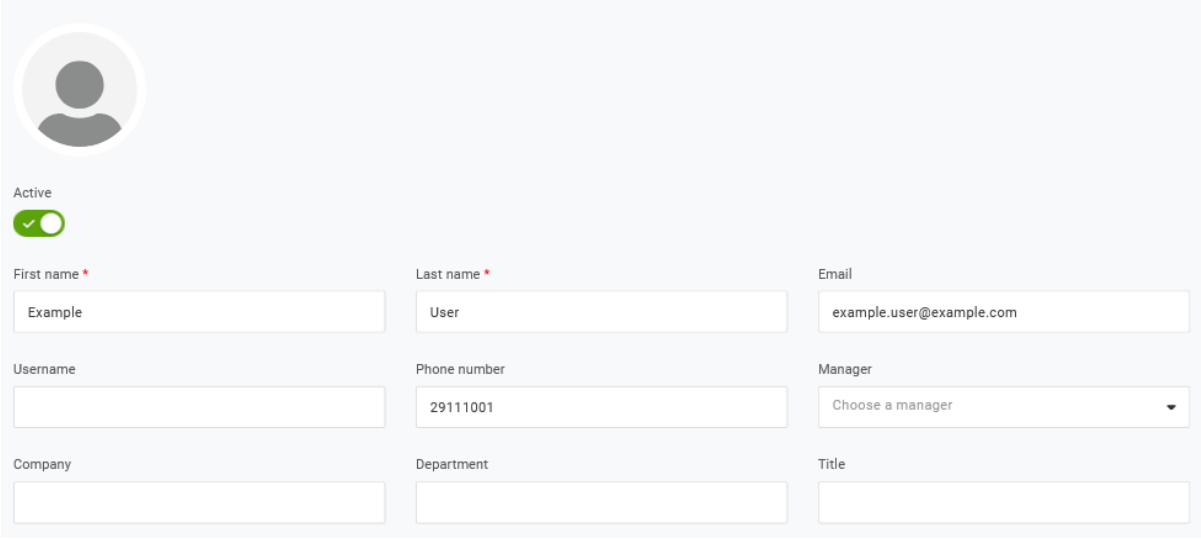
Получить сертификат для загрузки в OneLogin можно в разделе *Applications* > *SSO*: нажмите *View details* под текущим сертификатом.

3. Нажмите кнопку *Обновить*, чтобы сохранить эти настройки.

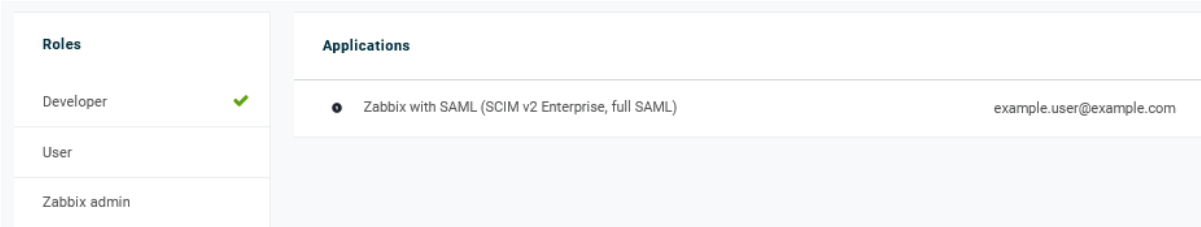
Подготовка пользователей SCIM

При включенной подготовке пользователей теперь можно добавлять/обновлять пользователей и их роли в OneLogin, после чего они будут немедленно подготовлены в Zabbix.

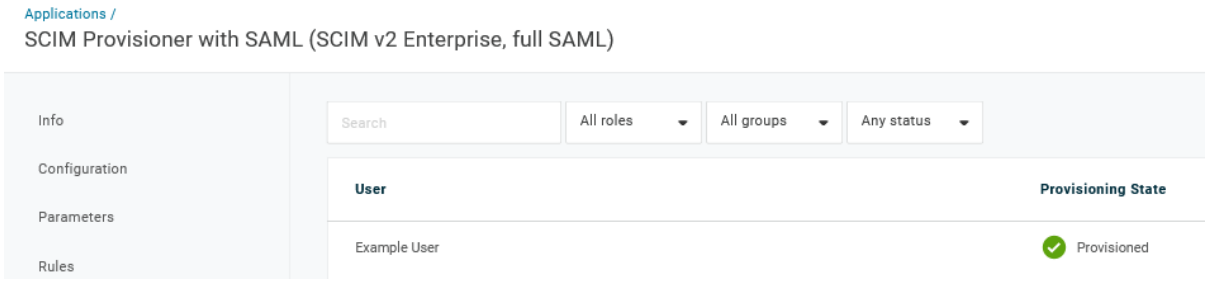
Например, можно создать нового пользователя:



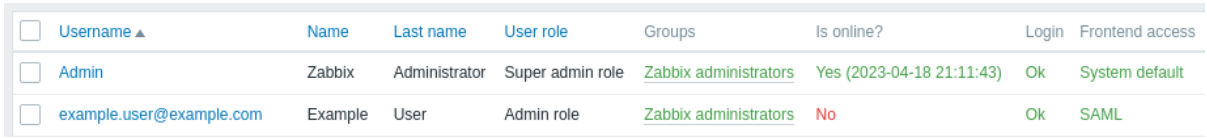
Добавьте его в роль пользователя и в приложение, которое будет выполнять подготовку пользователя:



При сохранении пользователя он будет подготовлен в Zabbix. В *Application* > *Users* можно проверить статус подготовки текущих пользователей приложения:



Если подготовка выполнена успешно, пользователь будет отображаться в списке пользователей Zabbix.



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

Настройка запланированных отчетов

Обзор

В этом разделе представлены инструкции по установке веб-сервиса Zabbix, а также настройке Zabbix, чтобы включить генерацию отчётов по расписанию.

Установка

Чтобы активировать создание отчётов по расписанию, должны быть установлены новый процесс веб-сервис Zabbix и браузер Google Chrome. Веб-сервис может быть установлен на том же компьютере, где установлен Zabbix сервер, или на другой машине. Браузер Google Chrome должен быть установлен на том же компьютере, где установлен веб-сервис.

Официальный пакет zabbix-web-service доступен в репозитории Zabbix. Браузер Google Chrome не включён в эти пакеты и должен быть установлен отдельно.

Чтобы скомпилировать веб-службу Zabbix из исходных кодов, см. Установка веб-службы Zabbix.

После установки запустите `zabbix_web_service` на машине, где установлена веб-служба:

```
zabbix_web_service
```

Настройка

Чтобы обеспечить корректное взаимодействие между всеми задействованными элементами, убедитесь, что параметры настройки сервера и настройки веб-интерфейса настроены правильно.

Сервер Zabbix

В конфигурационном файле сервера Zabbix необходимо обновить следующие параметры: `WebServiceURL` и `StartReportWriters`.

WebServiceURL

Этот параметр необходим для включения взаимодействия с веб-службой.
URL должен иметь формат `http[s]://host:port/report`.

- По умолчанию веб-служба прослушивает порт 10053.
  Другой порт можно указать в конфигурационном файле веб-службы.
- Указание пути `/report` обязательно (путь задан жестко и не может быть изменен).

Пример:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Настоятельно рекомендуется настроить шифрование между сервером Zabbix и веб-службой Zabbix с использованием сертификатов.
По умолчанию данные, передаваемые между сервером Zabbix и веб-службой Zabbix, не шифруются, что может привести к несанкционированному доступу.

StartReportWriters

Этот параметр определяет количество запускаемых процессов формирования отчетов.
Если параметр не задан или равен 0, формирование отчетов отключено.
В зависимости от количества и частоты требуемых отчетов можно включить от 1 до 100 процессов формирования отчетов.

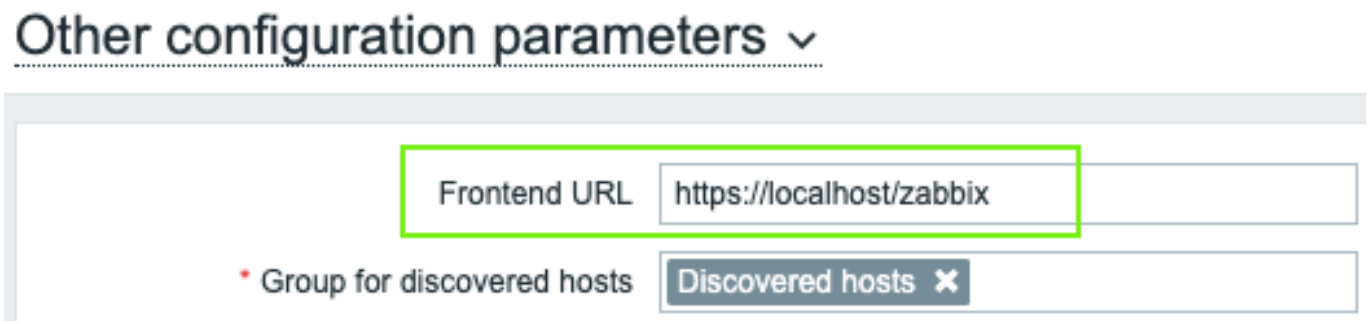
Пример:

```
StartReportWriters=3
```

Веб-интерфейс Zabbix

Чтобы разрешить коммуникации между веб-интерфейсом Zabbix и веб-сервисом Zabbix, должен быть выставлен параметр *URL веб-интерфейса (Frontend URL)* :

- Перейдите к разделу меню веб-интерфейса *Администрирование → Общие → Другое (Administration → General → Other)*
- Укажите полный URL веб-интерфейса Zabbix в параметре *URL веб-интерфейса (Frontend URL)*.



Note:
Чтобы убедиться, что всё работает правильно, после завершения процедуры первоначальной настройки Вы можете захотеть настроить и отослать тестовый отчёт.

Элементы данных

Этот раздел содержит справочную информацию и технические примечания об элементах данных, включая сведения о параметрах для конкретных ключей элементов данных, детали, зависящие от платформы, и примечания о специальных конфигурациях элементов данных.

Пожалуйста, используйте боковую панель, чтобы найти информацию, относящуюся к вашему случаю использования.

Параметры vm.memory.size

Обзор

В этом разделе представлены некоторые подробности относительно элемента данных vm.memory.size[<режим>] агента Zabbix.

Параметры

В этом элементе данных разрешены следующие параметры:

- **active** — память, используемая в данный момент или бывшвая в использовании совсем недавно, и поэтому ещё находится в RAM.
- **anon** — память, не связанная с файлами (повторное чтение из них невозможно).
- **available** — доступная память, вычисляется по-разному в зависимости от платформы (смотрите таблицу ниже).
- **buffers** — кэш для таких вещей, как метаданные файловой системы.
- **cached** — кэш для различных вещей.
- **exec** — исполняемый код, в основном из (программ) файлов.
- **file** — кэш содержимого недавно использованных файлов.
- **free** — память, которая доступна без каких-либо проблем любому объекту, запрашивающему память.
- **inactive** — память, помеченная как неиспользуемая.
- **pavailable** — «available» память в процентах по отношению к «total» (рассчитывается как `available/total*100`).
- **pinned** — то же, что и «wired».
- **pused** — «used» память в процентах по отношению к «total» (рассчитывается как `used/total*100`).
- **shared** — память, которая может быть доступна сразу нескольким процессам.

- **slab** — общий объем памяти, которая используется ядром для кэширования структур данных для собственного использования.
- **total** — общий объем доступной физической памяти.
- **used** — используемая память, вычисляется по-разному в зависимости от платформы (смотри таблицу ниже).
- **wired** — память, помеченная всегда оставаться в RAM. Она не может быть перемещена на диск.

Warning:
Некоторые из этих параметров работают только для конкретных платформ и могут быть недоступны на вашей платформе. См. элементы данных Zabbix агента.

Вычисления **available** и **used** в зависимости от платформы:

Платформа	**”available”**	**”used”**
AIX	free + cached	реальное использование памяти
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (также перенесено для 3.10 на RHEL 7)	/proc/meminfo, см. описание ”MemAvailable” в документации ядра Linux [en]. Обратите внимание, что free + buffers + cached больше не равняется ’available’, так как не весь кэш страницы может быть освобождён и в расчетах используется минимальный объем свободной памяти, зарезервированной системой (low watermark).	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
Сумма *vm.memory.size[used]* и *vm.memory.size[available]* не обязательно равна общему количеству памяти. Например, для FreeBSD:
 * Активная, неактивная, wired, кэшируемая памяти считаются использованными, так как содержат некоторую полезную информацию.
 * В то же время неактивная, кэшируемая, свободная памяти считаются доступными, так как такая память может быть незамедлительно освобождена процессу, который запросил больше памяти.
Таким образом, неактивная память помечается и как занятая, так и как свободная одновременно. Из-за этого элемент данных *vm.memory.size[used]* задумывался исключительно в информационных целях, тогда как элемент данных *vm.memory.size[available]* предназначен для использования в триггерах.

Смотрите также

1. Дополнительная информация о вычислении памяти на разных ОС [en]

Минимальный уровень разрешений для элементов данных агента Windows

Обзор

Когда системы мониторинга используют агента, хорошей практикой будет получение метрик с хоста, на котором установлен этот агент. Чтобы использовать принцип минимальных привилегий, необходимо определить метрики, которые собираются агентом.

Таблица в этом документе позволит вам выбрать минимальные права, которые будут гарантировать корректную работу Zabbix агента.

Если для работы агента выбран пользователь, отличный от «LocalSystem», то для работы агента в качестве службы Windows новый пользователь должен иметь права «Вход в качестве службы» из «Групповая политика→Назначение прав пользователя» и право создавать, записывать и удалять файл журнала Zabbix агента. Пользователь Active Directory должен быть добавлен в группу «Performance Monitor Users».

Note:
При работе с правами агента основываясь на группе «Минимально технически приемлемо», требуется предварительное предоставление прав на объекты мониторинга.

Поддерживаемые на Windows стандартные элементы данных агента

Ключ элемента данных	Группа пользователей	
	Рекомендуется	Минимально технически приемлемо (функциональность ограничена)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Ключи элементов данных, специфичные для Windows

Ключ элемента данных	Группа пользователей	
	Рекомендуется	Минимально технически приемлемо (функциональность ограничена)
eventlog	Читатели журнала событий	Гости
net.if.list	Гости	Гости
perf_counter	Пользователи Performance Monitor	Пользователи Performance Monitor
proc_info	Администраторы	Гости
service.discovery	Гости	Гости
service.info	Гости	Гости
services	Гости	Гости
wmi.get	Администраторы	Гости
vm.vmemory.size	Гости	Гости

Кодировка возвращаемых значений

Сервер Zabbix ожидает, что каждое возвращаемое текстовое значение будет в кодировке UTF8. Это относится к любому типу проверок: агент Zabbix, SSH, Telnet и т. д.

Разные контролируемые системы/устройства и проверки могут возвращать в значении не-ASCII символы. В таких случаях почти все возможные ключи **zabbix** содержат дополнительный параметр ключа элемента данных - **<encoding>**. Этот параметр ключа необязателен, но его следует указывать, если возвращаемое значение не в кодировке UTF8 и содержит не-ASCII символы. В противном случае результат может быть неожиданным и непредсказуемым.

Ниже приводится описание поведения с разными бэкендами баз данных в таких случаях.

MySQL

Если значение содержит не-ASCII символ в кодировке, отличной от UTF8, этот символ и последующие будут отброшены, когда база данных сохранит это значение. В `zabbix_server.log` не будут записаны предупреждения.
 Актуально как минимум для MySQL версии 5.1.61.

PostgreSQL

Если значение содержит не-ASCII символ в кодировке, отличной от UTF8, это приведет к сбою SQL-запроса (PGRES_FATAL_ERROR:ERRO
invalid byte sequence for encoding), и данные не будут сохранены. Соответствующее предупреждающее сообщение будет записано в `zabbix_server.log`.
 Актуально как минимум для PostgreSQL версии 9.1.3.

Поддержка больших файлов

Поддержка больших файлов, часто сокращаемая до LFS, — это термин, обозначающий возможность работы с файлами размером более 2 GB в 32-разрядных операционных системах. Поддержка больших файлов влияет как минимум на мониторинг файлов журнала и все элементы данных vfs.file.*. Поддержка больших файлов зависит от возможностей системы на момент компиляции Zabbix, но полностью отключена в 32-разрядной Solaris из-за ее несовместимости с procfs и swapctl.

Датчик

Обзор

Каждый контроллер датчика имеет свой собственный каталог в дереве sysfs `/sys/devices`. Чтобы найти все контроллеры датчиков, проще перейти по символическим ссылкам на устройства из `/sys/class/hwmon/hwmon*`, где * — это конкретный номер (0,1,2,...).

Показания датчиков находятся либо в папке `/sys/class/hwmon/hwmon*/` для виртуальных устройств, либо в папке `/sys/class/hwmon/hwmon*/device` для невиртуальных устройств. Файл с именем `name`, расположенный в папках `hwmon*` или `hwmon*/device`, содержит имя чипа, которое соответствует имени драйвера ядра, используемого контроллером датчика.

В каждом файле имеется только одно значение показания датчика. Общая схема именования файлов, которые содержат показания датчиков, внутри любой из вышеупомянутых папок: <тип><номер>_<элемент>, где

- **тип** — для чипов датчиков «in» (напряжение), «temp» (температура), «fan» (скорость вентилятора) и т.д.
- **элемент** — «input» (измеренное значение), «max» (верхний порог), «min» (нижний порог) и т.д.

- **номер** — всегда используется для экземпляров, которые могут присутствовать более одного раза (обычно начинается с 1, за исключением датчиков напряжения, которые начинаются с 0). Если файлы не относятся к конкретным экземплярам, они имеют простое наименование без номера.

Информацию о датчиках, доступных на хосте, можно получить с помощью инструментов **sensors-detect** и **sensors** (пакет lm-sensors [en]). **Sensors-detect** помогает определить, какие модули необходимы для доступных датчиков. Когда модули загружены, можно использовать утилиту **sensors**, чтобы отобразить показания со всех контроллеров. Маркировка показаний датчиков, используемая данной утилитой, может отличаться от общепринятой схемы именования (< >< >_< >):

- если имеется файл с именем < >< >_ , то вместо имени < >< >_< > будет использоваться метка внутри этого файла;
- если файл < >< >_ отсутствует, то для подстановки имени утилита выполняет поиск внутри `/etc/sensors.conf` (также может быть `/etc/sensors3.conf` или другой).

Эта маркировка позволяет пользователю определить тип используемого оборудования. Если нет ни файла < >< >_ , ни метки внутри файла конфигурации, то тип оборудования может быть определён по атрибуту имени (`hwmon*/device/name`). Реальные имена датчиков, которые воспринимает `zabbix_agent`, можно получить, запустив утилиту **sensors** с параметром «`-u`» (`sensors -u`).

В утилите **sensors** доступные датчики разделены по типу шины (ISA adapter, PCI adapter, SPI adapter, Virtual device, ACPI interface, HID adapter).

Получение идентификаторов датчиков

Метки датчиков не всегда могут быть использованы в том виде, как они выводятся утилитой `sensors`, т.к. у разных производителей контроллеров именование меток может отличаться. Например, вывод утилиты `sensors` может содержать следующие строки:

```
sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

Из них только одно отображаемое имя напрямую соответствует внутреннему идентификатору:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Попытки использовать остальные отображаемые метки (такие как `Vcore` или `+12V`) работать не будут.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Чтобы найти внутренний идентификатор датчика, который воспринимает Zabbix, запустите `sensors -u`. В полученном выводе можно увидеть следующее:

```
sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

Таким образом, метка `Vcore` соответствует идентификатору `in1`, а `+12V` соответствует `in4`. Согласно спецификации [en] hwmon sysfs ядра, это напряжения на выводах микросхемы и, в общем случае, может понадобиться масштабирование.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

В Zabbix можно получать не только показания напряжения (in), но также тока (curr), температуры (temp) и скорости вращения вентилятора (fan).

Совместимость

Устаревшие дистрибутивы

Показания датчиков извлекаются из папки /proc/sys/dev/sensors:

- **устройство** — имя устройства (если <режим> используется, это регулярное выражение)
- **датчик** — имя датчика (если используется <режим>, это регулярное выражение)
- **режим** — возможные значения: avg, max, min (если этот параметр не указан, устройство и сенсор обрабатываются буквально)

Пример ключа: sensor[w83781d-i2c-0-2d,temp1].

Текущие дистрибутивы

Показания датчиков извлекаются из папки /sys/class/hwmon:

- **устройство** — имя устройства (не регулярное выражение). Имя устройства может быть фактическим именем устройства (например, 0000:00:18.3) или именем, полученным при использовании утилиты sensors (например, k8temp-pci-00c3). Выбор того, какое из этих имён использовать, оставлен за пользователем.
- **датчик** — имя датчика (не регулярное выражение).
- **режим** — возможные значения: avg, max, min (если этот параметр не указан, устройство и сенсор обрабатываются буквально).

Примеры ключей:

sensor[k8temp-pci-00c3,temp,max] или sensor[0000:00:18.3,temp1].

sensor[smsc47b397-isa-0880,in,avg] или sensor[smsc47b397.2176,in1].

Примечания к параметру memtype в элементах данных proc.mem

Обзор

Параметр **memtype** поддерживается на платформах Linux, AIX, FreeBSD и Solaris.

На всех этих платформах поддерживаются три распространённых значения memtype: pmem, rss и vsize. Кроме того, на некоторых платформах поддерживаются специфичные для платформы значения memtype.

AIX

В таблице приведены значения, поддерживаемые для параметра memtype в AIX.

Поддерживаемое значение	Описание	Источник в структуре procentry64	Пытается обеспечить совместимость с
vsize [1]	Размер виртуальной памяти	pi_size	
pmem	Процент реальной памяти	pi_prm	ps -o pmem
rss	Размер резидентного набора	pi_trss + pi_drss	ps -o rssize
size	Размер процесса (код + данные)	pi_dvm	столбец SIZE команды ps gvw
dsize	Размер данных	pi_dsize	
tsize	Размер текста (кода)	pi_tsize	столбец TSIZ команды ps gvw
sdsize	Размер данных из общей библиотеки	pi_sdsize	
drss	Размер резидентного набора данных	pi_drss	
trss	Размер резидентного набора текста	pi_trss	

Примечания для AIX:

1. При выборе параметров для ключа элемента данных proc.mem[] в AIX старайтесь указывать узкие критерии выбора процессов. В противном случае существует риск включения нежелательных процессов в результат proc.mem[].

Пример:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Этот пример показывает, что указание только параметра командной строки (регулярного выражения для сопоставления) приводит к учёту самого агента Zabbix, что, вероятно, не является желаемым результатом.

2. Не используйте `ps -ef` для просмотра процессов — эта команда показывает только процессы, не относящиеся к ядру. Используйте `ps -Af`, чтобы увидеть все процессы, доступные агенту Zabbix.
3. Рассмотрим на примере `topasrec`, как агент Zabbix выбирает процессы с помощью `proc.mem[]`.

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

`proc.mem[]` имеет следующие аргументы:

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

где: - Первый критерий - имя процесса (аргумент `<name>`). В нашем примере агент Zabbix увидит его как `topasrec`. Для соответствия необходимо указать `topasrec` или оставить поле пустым. - Второй критерий - имя пользователя (аргумент `<user>`). Для соответствия необходимо указать `root` или оставить поле пустым. - Третий критерий, используемый при выборе процесса, - аргумент `<cmdline>`. Агент Zabbix увидит его значение как `/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023`. Для соответствия необходимо указать регулярное выражение, соответствующее этой строке, или оставить поле пустым.

Аргументы `<mode>` и `<memtype>` применяются после использования трех указанных выше критериев.

FreeBSD

В таблице приведены значения, поддерживаемые для параметра `memtype` в FreeBSD.

Поддерживаемое значение	Описание	Источник в структуре kinfo_proc	Попытка совместимости с
`vsize`	Размер виртуальной памяти	`kp_eproc.e_vm.vm_map_size` или `ki_size`	`vsz`
`pmem`	Процент реальной памяти	вычисляется на основе `rss`	`ps -o pmem`
`rss`	Размер резидентного набора	`kp_eproc.e_vm.vm_rssize` или `ki_rssize`	`rss`
`size` [1]	Размер процесса (код + данные + стек)	`tsize + dsize + ssize`	
`tsize`	Размер текста (кода)	`kp_eproc.e_vm.vm_tsize` или `ki_tsize`	`tsiz`
`dsize`	Размер данных	`kp_eproc.e_vm.vm_dsize` или `ki_dsize`	`dsiz`
`ssize`	Размер стека	`kp_eproc.e_vm.vm_ssize` или `ki_ssize`	`ssiz`

Linux

Поддерживаемые значения параметра `memtype` в Linux приведены в таблице.

Поддерживаемое значение	Описание	Источник в файле /proc/<pid>/status
`vsize` [1]	Размер виртуальной памяти	`VmSize`
`pmem`	Процент физической памяти	`(VmRSS/total_memory) * 100`
`rss`	Размер резидентного набора	`VmRSS`
`data`	Размер сегмента данных	`VmData`
`exe`	Размер сегмента кода	`VmExe`
`hwm`	Пиковый размер резидентного набора	`VmHWM`
`lck`	Размер заблокированной памяти	`VmLck`
`lib`	Размер общих библиотек	`VmLib`
`peak`	Пиковый размер виртуальной памяти	`VmPeak`
`pin`	Размер закреплённых страниц	`VmPin`
`pte`	Размер записей таблицы страниц	`VmPTE`
`size`	Размер сегментов кода, данных и стека процесса	`VmExe + VmData + VmStk`
`stk`	Размер сегмента стека	`VmStk`

Поддерживаемое значение	Описание	Источник в файле /proc/<pid>/status
swap	Размер используемого пространства подкачки	VmSwap

Примечания для Linux:

1. Не все значения memtype поддерживаются старыми ядрами Linux. Например, ядра Linux 2.4 не поддерживают значения hwm, pin, peak, pte и swap.
2. Мы заметили, что самоконтроль процесса активной проверки агента Zabbix с помощью proc.mem[...,...,...,...,data] показывает значение, на 4 кБ превышающее значение, указанное в строке VmData файла /proc/<pid>/status агента. Во время самоконтроля сегмент данных агента увеличивается на 4 кБ, а затем возвращается к прежнему размеру.

Solaris

Поддерживаемые значения параметра memtype в Solaris приведены в таблице.

Поддерживаемое значение	Описание	Источник в структуре psinfo	Совместимость с
vsize [1]	Размер образа процесса	pr_size	ps -o vsz
pmem	Процент реальной памяти	pr_pctmem	ps -o pmem
rss	Размер резидентного набора Может быть занижен (см. описание rss в man ps).	pr_rssize	ps -o rss

Примечания

[1] Значение по умолчанию.

Примечания по выбору процессов в элементах данных proc.mem и proc.num

Процессы, изменяющие свою командную строку

Некоторые программы используют изменение своей командной строки как метод отображения своей текущей активности. Пользователь может увидеть эту активность, запуская команды ps и top. Примеры таких программ: *PostgreSQL*, *Sendmail*, *Zabbix*.

Давайте рассмотрим пример из Linux. Предположим, что мы хотим наблюдать количество процессов Zabbix агента.

Команда ps отобразит интересующие процессы следующим образом:

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

Выбор процессов по имени и пользователю выполняет задачу:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Теперь давайте переименуем исполняемый файл zabbix_agentd в zabbix_agentd_30 и перезапустим его.

Теперь ps отображает:

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
```

```
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

Теперь выбор процессов по имени и пользователю выдаёт неверный результат:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Почему простое переименование исполняемого файла на более длинное имя приводит к совершенно другому результату?

Zabbix agent начинает с проверки имени процесса. Открывается файл /proc/<pid>/status и проверяется строка Name. В нашем случае строки Name это:

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

Имя процесса в файле status обрезается до 15 символов.

Аналогичный результат можно увидеть при помощи команды ps:

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

Очевидно, что этот вывод не идентичен нашему значению zabbix_agentd_30 в параметре name ключа proc.num[]. Будучи не в состоянии найти совпадение по имени процесса в файле status, Zabbix агент обращается к файлу /proc/<pid>/cmdline.

То, как агент просматривает файл «cmdline», может быть проиллюстрировано при помощи выполнения команды:

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

В нашем случае файлы /proc/<pid>/cmdline содержат невидимые, непечатаемые нулевые байты, которые используются для завершения строки в языке C. В этом примере нулевые байты отображаются как «<NUL>».

Zabbix агент проверяет «cmdline» основного процесса и извлекает zabbix_agentd_30, которое соответствует значению zabbix_agentd_30 нашего параметра name. Таким образом, основной процесс учитывается элементом данных proc.num[zabbix_agentd_30,zabbix].

При проверке следующего процесса агент из файла cmdline получает zabbix_agentd_30: collector [idle 1 sec], и это не соответствует нашему значению zabbix_agentd_30 в параметре name. Таким образом, будет сосчитан только основной процесс, который не меняет свою командную строку. Остальные процессы агента модифицируют свои командные строки и игнорируются.

Этот пример показывает, что в данном случае параметр name нельзя использовать в proc.mem[] и proc.num[] для выбора процессов.

Note:
Для элемента данных `proc.get[]`, когда агент Zabbix проверяет «cmdline» на предмет имени процесса, он будет использовать только часть имени, начиная с последнего слеша и до первого пробела или знака двоеточия. Имя процесса, полученное из файла cmdline, будет использоваться только в том случае, если его начало полностью совпадает с сокращённым именем процесса в файле `status`. Алгоритм одинаков как для имени процесса в фильтре, так и для вывода JSON.

Использование параметра `cmdline` с надлежащим регулярным выражением даст правильный результат:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Будьте осторожны при использовании элементов данных `proc.get[]`, `proc.mem[]` и `proc.num[]` при наблюдении за программами, которые модифицируют свои командные строки.

Перед тем как поместить параметры `name` и `cmdline` в элементы данных `proc.get[]`, `proc.mem[]` и `proc.num[]`, вы, возможно, захотите протестировать эти параметры, используя элемент данных `proc.num[]` и команду `ps`.

Потоки ядра Linux

Нельзя выбрать потоки при помощи параметров `cmdline` в элементах данных `proc.get[]`, `proc.mem[]` и `proc.num[]`

Давайте возьмем в качестве примера один из потоков ядра:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

Его можно выбрать при помощи параметра :

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Но выбор при помощи параметра `cmdline` не работает:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

Причина такого поведения кроется в том, что Zabbix агент берёт регулярное выражение, которое указано в параметре `cmdline`, и применяет его к содержимому относящегося к процессу файла `/proc/<pid>/cmdline`. В случае потоков ядра, их файлы `/proc/<pid>/cmdline` пустые. Поэтому параметр `cmdline` никогда не совпадёт.

Подсчёт потоков в элементах данных `proc.mem[]` и `proc.num[]`

Потоки ядра Linux подсчитываются элементом данных `proc.num[]`, но не сообщают информацию о памяти в элементе данных `proc.mem[]` item. Например:

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

Но что случится, если имеется пользовательский процесс с таким же именем как и у потока ядра? Тогда это будет выглядеть примерно так:

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

`proc.num[]` подсчитал как поток ядра, так и пользовательский процесс. `proc.mem[]` сообщил информацию о памяти только по пользовательскому процессу и вычислил память потока ядра как будто оно было нулевым. Это отличается от случая выше, когда сообщалось ZBX_NOTSUPPORTED.

Будьте внимательны при использовании элементов данных `proc.mem[]` и `proc.num[]`, если случится, что имя процесса совпадёт с одним из потоков.

Перед добавлением параметров в элементы данных `proc.mem[]` и `proc.num[]` вы, возможно, захотите протестировать эти параметры, используя элемент данных `proc.num[]` и команду `ps`.

Детали реализации проверок net.tcp.service и net.udp.service

Реализация проверок net.tcp.service и net.udp.service подробно описана на этой странице для различных служб, указанных в параметре service.

Параметры элемента данных net.tcp.service

ftp

Создаёт TCP-соединение и ожидает, что первые 4 символа ответа будут «220 », после чего посылает «QUIT\r\n». По умолчанию используется порт 21, если он не указан.

http

Создаёт TCP-соединение без ожидания и отсылки чего-либо. По умолчанию используется порт 80, если он не указан.

https

Использует (и работает только с) libcurl, не проверяет подлинность сертификата, не проверяет имя хоста в SSL сертификате, только извлекает заголовок ответа (запрос HEAD). По умолчанию используется порт 443, если он не указан.

imap

Создаёт TCP-соединение и ожидает, что первые 4 символа ответа будут «* OK», после чего посылает «a1 LOGOUT\r\n». По умолчанию используется порт 143, если он не указан.

ldap

Открывает соединение на сервер LDAP и выполняет операцию поиска (LDAP search) с фильтром, выставленным в (objectClass=*). Ожидает успешного извлечения первого атрибута у первого элемента. По умолчанию используется порт 389, если он не указан.

nntp

Создаёт TCP-соединение и ожидает, что первые 3 символа ответа будут «200» или «201», после чего посылает «QUIT\r\n». По умолчанию используется порт 119, если он не указан.

pop

Создаёт TCP-соединение и ожидает, что первые 3 символа ответа будут «+OK», после чего посылает «QUIT\r\n». По умолчанию используется порт 110, если он не указан.

smtp

Создаёт TCP-соединение и ожидает, что первые 3 символа ответа будут «220», после которых идёт пробел, конец строки или минус. Строки, содержащие минус, относятся к многострочному ответу, и ответ будет заново читаться до тех пор, пока не будет получена строка без минуса. После этого отсылается «QUIT\r\n». По умолчанию используется порт 25, если он не указан.

ssh

Создаёт TCP-соединение. Если соединение было установлено, обе стороны обмениваются строками идентификации (SSH-major.minor-XXXX), где major и minor являются версиями протокола, а XXXX - строкой. Zabbix проверяет, найдена ли соответствующая спецификации строка, и затем отсылает назад строку «SSH-major.minor-zabbix_agent\r\n» или «0\n» при несовпаднии. По умолчанию используется порт 22, если он не указан.

tcp

Создаёт TCP-соединение без ожидания и отсылки чего-либо. В отличие от остальных проверок, требует указания параметра «порт».

telnet

Создаёт TCP-соединение и ожидает приглашение на ввод имени пользователя («:» в конце строки). По умолчанию используется порт 23, если он не указан.

Параметры элемента данных net.udp.service

ntp

Отсылает по UDP пакет SNTP и проверяет ответ в соответствии с RFC 4330, раздел 5 [en]. По умолчанию используется порт 123, если он не указан.

Параметры proc.get

Обзор

Элемент данных **proc.get**[<имя>,<пользователь>,<cmdline>,<режим>] поддерживается в Linux, Windows, FreeBSD, OpenBSD и NetBSD.

Список параметров процесса, возвращаемых элементом данных, зависит от операционной системы и значения аргумента «режим».

Linux

В Linux для каждого режима возвращаются следующие параметры процесса:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: имя процесса
ppid: родительский PID	ppid: родительский PID	processes: количество процессов
name: имя процесса	name: имя процесса	vsize: размер виртуальной памяти
cmdline: команда с аргументами	user: пользователь (реальный), от имени которого выполняется процесс	pmem: процент реальной памяти
user: пользователь (реальный), от имени которого выполняется процесс	group: группа (реальная), от имени которой выполняется процесс	rss: размер резидентного набора
group: группа (реальная), от имени которой выполняется процесс	uid: идентификатор пользователя	data: размер сегмента данных
uid: идентификатор пользователя	gid: идентификатор группы, от имени которой выполняется процесс	exe: размер сегмента кода
gid: идентификатор группы, от имени которой выполняется процесс	tid: идентификатор потока	lib: размер общих библиотек
vsize: размер виртуальной памяти	tname: имя потока	lck: размер заблокированной памяти
pmem: процент реальной памяти	cputime_user: общее время ЦП, затраченное в пользовательском режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)	pin: размер закрепленных страниц
rss: размер резидентного набора	cputime_system: общее время ЦП, затраченное в системном режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)	pte: размер элементов таблицы страниц
data: размер сегмента данных	state: состояние потока	size: размер сегментов кода, данных и стека процесса
exe: размер сегмента кода	ctx_switches: количество переключений контекста	stk: размер сегмента стека
hwm: пиковый размер резидентного набора	page_faults: количество ошибок страниц	swap: размер используемого пространства подкачки
lck: размер заблокированной памяти		cputime_user: общее время ЦП, затраченное в пользовательском режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)

mode=process	mode=thread	mode=summary
lib: размер общих библиотек		cputime_system: общее время ЦП, затраченное в системном режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)
peak: пиковый размер виртуальной памяти		ctx_switches: количество переключений контекста
pin: размер закрепленных страниц		threads: количество потоков
pte: размер элементов таблицы страниц		page_faults: количество ошибок страниц
size: размер сегментов кода, данных и стека процесса		pss: пропорциональный размер набора памяти
stk: размер сегмента стека		
swap: размер используемого пространства подкачки		
cputime_user: общее время ЦП, затраченное в пользовательском режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)		
cputime_system: общее время ЦП, затраченное в системном режиме (значение выражено в тактах часов — разделите его на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)		
state: состояние процесса (прозрачно полученное из procfs, полная форма)		
ctx_switches: количество переключений контекста		
threads: количество потоков		
page_faults: количество ошибок страниц		
pss: пропорциональный размер набора памяти		

ОС на основе BSD

На FreeBSD, OpenBSD и NetBSD для каждого режима возвращаются следующие параметры процесса:

режим=process	режим=thread	режим=summary
pid: идентификатор процесса	pid: идентификатор процесса	name: имя процесса
ppid: идентификатор родительского процесса	ppid: идентификатор родительского процесса	processes: количество процессов
jid: идентификатор песочницы (jail, только FreeBSD)	jid: идентификатор песочницы (jail, только FreeBSD)	vsize: размер виртуальной памяти
jname: имя песочницы (jail, только FreeBSD)	jname: имя песочницы (jail, только FreeBSD)	pmem: процент реальной памяти (только FreeBSD)
name: имя процесса	name: имя процесса	rss: размер резидентного набора
cmdline: командная строка с аргументами	user: пользователь (реальный), от имени которого работает процесс	size: размер процесса (код + данные + стек)
user: пользователь (реальный), от имени которого работает процесс	группа (реальная), от имени которой работает процесс	tsize: размер текста (код)

режим=process	режим=thread	режим=summary
group: группа (реальная), от имени которой работает процесс	uid: идентификатор пользователя	dsize: размер данных
uid: идентификатор пользователя	gid: идентификатор группы, от имени которой работает процесс	ssize: размер стека
gid: идентификатор группы, от имени которой работает процесс	tid: идентификатор потока (thread ID)	cputime_user: общее время работы CPU в режиме пользователя (в секундах)
vsize: размер виртуальной памяти	tname: имя потока	cputime_system: общее время работы CPU в секундах в режиме системы (в секундах)
pmem: процент реальной памяти (только FreeBSD)	cputime_user: общее время работы CPU в режиме пользователя (в секундах)	ctx_switches: количество переключений контекста
rss: размер резидентного набора	cputime_system: общее время работы CPU в режиме системы (в секундах)	threads: количество потоков (не поддерживается для NetBSD)
size: размер процесса (код + данные + стек)	state: состояние потока	stk: размер сегмента стека
tsize: размер текста (код)	ctx_switches: количество переключений контекста	page_faults: количество ошибок страниц
dsize: размер данных	io_read_op: количество раз, когда системе пришлось выполнить ввод	fds: количество файловых дескрипторов (только OpenBSD)
ssize: размер стека	io_write_op: количество раз, когда системе пришлось выполнить вывод	swap: размер используемого пространства подкачки
cputime_user: общее время работы CPU в режиме пользователя (в секундах)		io_read_op: количество раз, когда системе пришлось выполнить ввод
cputime_system: общее время работы CPU в режиме системы (в секундах)		io_write_op: количество раз, когда системе пришлось выполнить вывод
state: состояние процесса (disk sleep/running/sleeping/tracing stop/zombie/other)		
ctx_switches: количество переключений контекста		
threads: количество потоков (не поддерживается для NetBSD)		
page_faults: количество ошибок страниц		
fds: количество файловых дескрипторов (только OpenBSD)		
swap: размер используемого пространства подкачки		
io_read_op: количество раз, когда системе пришлось выполнить ввод		
io_write_op: количество раз, когда системе пришлось выполнить вывод		

Windows

В Windows для каждого режима возвращаются следующие параметры процесса:

режим=process	режим=thread	режим=summary
pid: идентификатор процесса	pid: идентификатор процесса	name: имя процесса
ppid: идентификатор родительского процесса	ppid: идентификатор родительского процесса	processes: количество процессов
name: имя процесса	name: имя процесса	vmsize: размер виртуальной памяти
user: пользователь, под которым выполняется процесс	user: пользователь, под которым выполняется процесс	wkset: размер рабочего набора процесса

режим=process	режим=thread	режим=summary
sid: SID пользователя	sid: SID пользователя	cputime_user: общее время работы ЦП в режиме пользователя (в миллисекундах)
vmsize: размер виртуальной памяти	tid: идентификатор потока	cputime_system: общее время работы ЦП в режиме системы (в миллисекундах)
wkset: размер рабочего набора процесса		threads: количество потоков
cputime_user: общее время работы ЦП в режиме пользователя (в миллисекундах)		page_faults: количество ошибок страниц
cputime_system: общее время работы ЦП в режиме системы (в миллисекундах)		handles: количество дескрипторов
threads: количество потоков		io_read_b: прочитано байтов ввода-вывода
page_faults: количество ошибок страниц		io_write_b: записано байтов ввода-вывода
handles: количество дескрипторов		io_read_op: операций чтения ввода-вывода
io_read_b: прочитано байтов ввода-вывода		io_write_op: операций записи ввода-вывода
io_write_b: записано байтов ввода-вывода		io_other_b: передано байтов ввода-вывода, кроме операций чтения и записи
io_read_op: операций чтения ввода-вывода		io_other_op: операций ввода-вывода, кроме операций чтения и записи
io_write_op: операций записи ввода-вывода		
io_other_b: передано байтов ввода-вывода, кроме операций чтения и записи		
io_other_op: операций ввода-вывода, кроме операций чтения и записи		

Настройки недоступного/не отвечающего интерфейса узла сети

Обзор

Несколько параметров конфигурации parameters определяют, как должен вести себя сервер Zabbix, когда проверка агента (Zabbix, SNMP, IPMI, JMX) завершается неудачей и интерфейс узла сети становится недоступным.

Недоступный интерфейс

Интерфейс узла сети считается недоступным после неудачной проверки (ошибка сети, тайм-аут) агентами Zabbix, SNMP, IPMI или JMX. Начиная с Zabbix 6.2.0, активные проверки агента Zabbix также влияют на доступность интерфейса. Когда активные проверки становятся недоступными, они учитываются в общем статусе доступности интерфейса агента.

С момента, когда интерфейс становится недоступным, **UnreachableDelay** определяет, как часто он перепроверяется с использованием одного из элементов данных (включая правила LLD). Эти повторные проверки выполняются опросчиками недоступных узлов (или IPMI-опросчиками для проверок IPMI). По умолчанию интервал между последовательными проверками доступности составляет 15 секунд.

> **Attention:**
> Проверки, выполняемые асинхронными опросчиками, не переносятся к опросчикам недоступных узлов.

В журнале сервера Zabbix недоступность обозначается сообщениями вроде:

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

Сообщения журнала указывают точный элемент данных, который не прошел проверку, и его тип (агент Zabbix).

Note:
Параметр *Timeout* также влияет на то, как рано интерфейс будет перепроверен во время недоступности. Если Timeout установлен в 20 секунд, а UnreachableDelay — в 30 секунд, следующая проверка будет через 50 секунд после первой попытки.

Параметр **UnreachablePeriod** определяет общую продолжительность периода недоступности. По умолчанию UnreachablePeriod равен 45 секундам. Это значение должно быть в несколько раз больше, чем UnreachableDelay, чтобы обеспечить многократную перепроверку интерфейса до того, как он будет помечен как недоступный.

Внутренний элемент данных zabbix[host,active_agent,available] позволяет отслеживать доступность активных проверок в сценариях недоступности.

Перевод интерфейса обратно в доступное состояние

По окончании периода недостижимости интерфейс опрашивается заново, со снижением приоритета элемента данных, который перевел интерфейс в недостижимое состояние. Если устанавливается связь с интерфейсом, который был недостижимым, мониторинг автоматически возвращается к норме:

```
resuming Zabbix agent checks on host "New host": connection restored
```

Note:
После того как интерфейс становится доступным, опрос всех элементов данных немедленно не осуществляется узлом сети по двум причинам:

- это может создать слишком большую нагрузку на узел сети,
- время восстановления интерфейса не всегда совпадает с плановым опросом элемента данных согласно заданному расписанию проверок.

Поэтому, когда интерфейс становится доступным, опрос элементов данных производится не сразу, а перепланируется в соответствии с расписанием их следующих проверок.

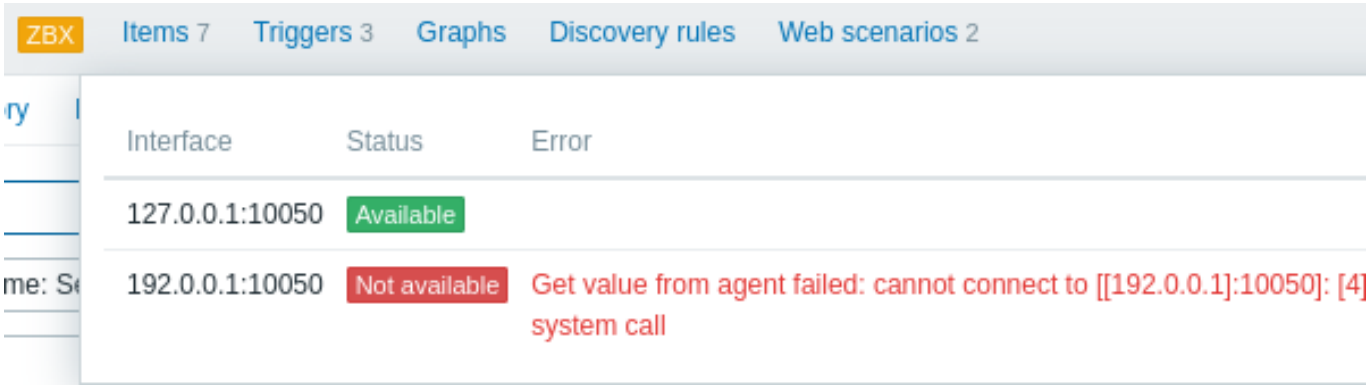
Недоступный интерфейс

После того как UnreachablePeriod завершился, а интерфейс так и не стал доступным, он считается недоступным.

В журнале сервера это обозначается записями такого рода:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

При этом в веб-интерфейсе иконка доступности узла сети меняет свой цвет с зеленого/серого на желтый/красный (при наведении указателя мыши на иконку доступности узла сети можно видеть информацию о недоступности интерфейса):



Параметр **UnavailableDelay** определяет, как часто интерфейс проверяется повторно в течение его недоступности.

По умолчанию — раз в 60 секунд (таким образом, в этом случае «temporarily disabling» из файла журнала выше будет означать деактивацию проверок сроком на одну минуту).

Когда соединение с интерфейсом будет восстановлено, мониторинг тоже автоматически вернётся к норме:

```
enabling Zabbix agent checks on host "New host": interface became available
```

Удаленный мониторинг статистики Zabbix

Обзор

Можно сделать некоторые внутренние метрики сервера Zabbix и прокси доступными удаленно для другого экземпляра Zabbix или стороннего инструмента. Это может быть полезно, чтобы поставщики услуг/поддержки могли удаленно контролировать серверы/прокси Zabbix своих клиентов, либо в организациях, где Zabbix не является основным инструментом мониторинга, чтобы внутренние метрики Zabbix могли отслеживаться сторонней системой в составе схемы централизованного мониторинга.

Внутренние статистические данные Zabbix доступны для настраиваемого набора адресов, указанных в новом параметре 'StatsAllowedIP' server/proxy. Запросы будут приниматься только с этих адресов.

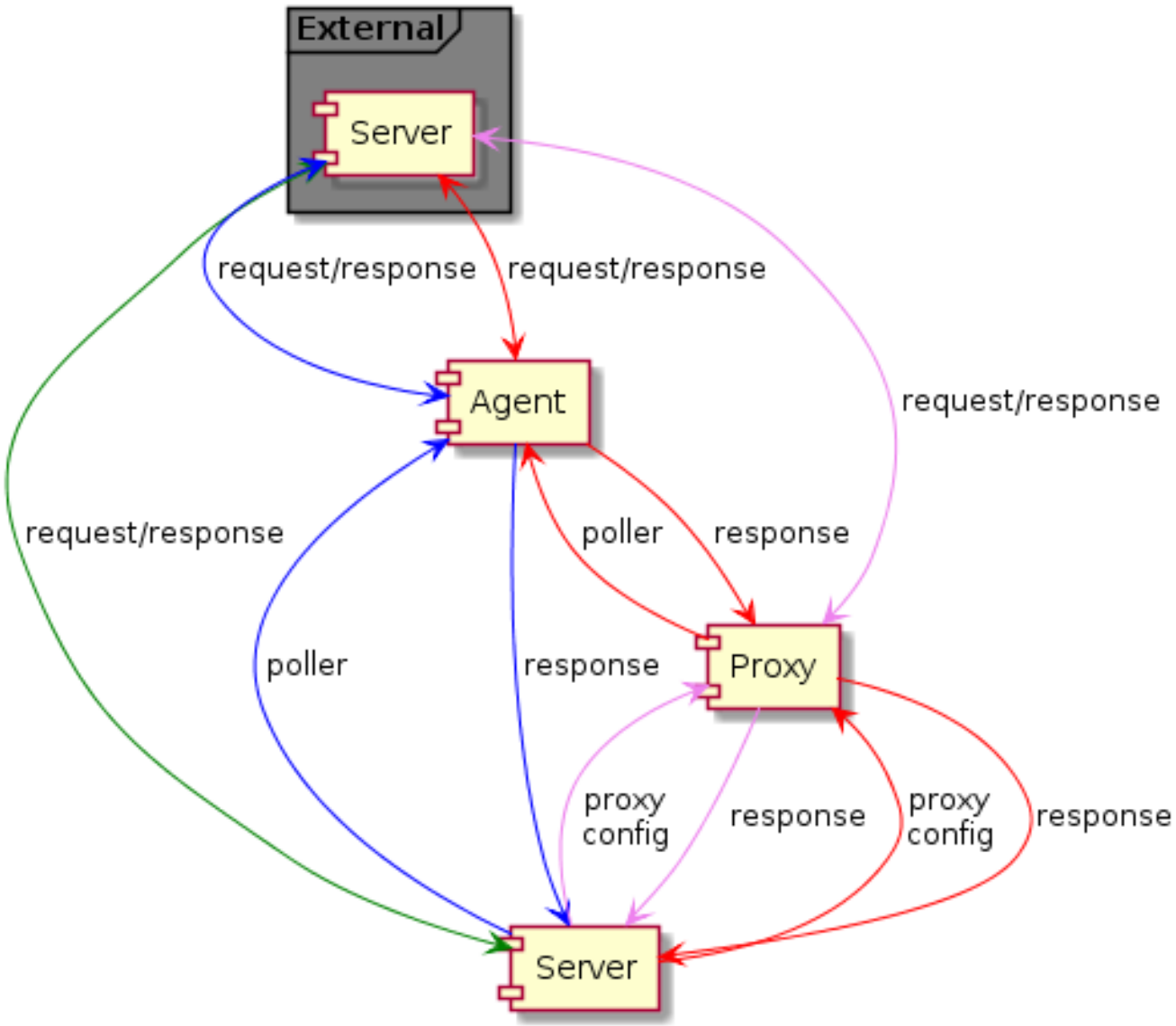
Элементы данных

Для настройки запроса внутренней статистики с другого экземпляра Zabbix вы можете использовать два элемента данных:

- Внутренний элемент данных `zabbix[stats,<ip>,<  >]` - для прямых удалённых запросов к Zabbix серверу/прокси. <ip> и <порт> используются для идентификации необходимого целевого экземпляра.
- Элемент данных агента `zabbix.stats[<ip>,<  >]` для удалённых запросов к Zabbix серверу/прокси через агента. <ip> и <порт> используются для идентификации необходимого целевого экземпляра.

Смотрите также: Внутренние элементы данных, Элементы данных Zabbix агента.

Следующая диаграмма иллюстрирует использование обоих элементов данных в зависимости от контекста.



- - Сервер → внешний экземпляр Zabbix (`zabbix[stats,<ip>,<  >]`)
- - Сервер → прокси → внешний экземпляр Zabbix (`zabbix[stats,<ip>,<  >]`)
- - Сервер → агент → внешний экземпляр Zabbix (`zabbix.stats[<ip>,<  >]`)

-  - Сервер → прокси → агент → внешний экземпляр Zabbix (`zabbix.stats[<ip>,<  >]`)

Чтобы убедиться, что целевой экземпляр разрешает запросы к себе со стороны внешнего экземпляра, укажите адрес внешнего экземпляра в параметре 'StatsAllowedIP' на целевом экземпляре.

Предоставляемые метрики

Элементы данных собирают статистику одним запросом и возвращают JSON, который является основой зависимых элементов данных и из которого они получают данные. Любым из двух элементов данных возвращаются следующие внутренние метрики:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (только сервер)
- `zabbix[preprocessing_queue]` (только сервер)
- `zabbix[process,<  >,<  >,<  >]` (статистика только на основе типов процессов)
- `zabbix[rcache,<  >,<  >]`
- `zabbix[requiredperformance]`
- `zabbix[triggers]` (только сервер)
- `zabbix[uptime]`
- `zabbix[vcache,buffer,<  >]` (только сервер)
- `zabbix[vcache,cache,<  >]`
- `zabbix[version]`
- `zabbix[vmware,buffer,<  >]`
- `zabbix[wcache,<  >,<  >]` (тип кэша «trends» только сервер)

Шаблоны

Для удалённого мониторинга внутренних метрик Zabbix сервера или прокси с внешнего экземпляра доступны шаблоны:

- Remote Zabbix server health
- Remote Zabbix proxy health

Обратите внимание, что для использования шаблонов удалённого мониторинга нескольких внешних экземпляров, для мониторинга каждого внешнего экземпляра потребуется отдельный узел сети.

Процесс траппер

Получение запросов внутренних метрик с внешнего экземпляра Zabbix обрабатывается процессом траппер, который проверяет запрос, собирает метрики, создаёт буфер данных JSON и отправляет назад подготовленный JSON, например, с сервера:

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
```

```
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
```

```
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```

Элементы данных внутренней очереди

Также существуют два других элемента данных, которые позволяют удаленно запрашивать статистику внутренней очереди в другом экземпляре Zabbix:

- внутренний элемент данных `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` - для прямых запросов к внутренней очереди удаленного сервера/прокси Zabbix
- элемент данных агента `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` - для запросов к внутренней очереди удаленного сервера/прокси Zabbix через агент

См. также: Внутренние элементы данных, Элементы данных агента Zabbix

Настройка Kerberos в Zabbix

Обзор

В Zabbix аутентификацию Kerberos можно использовать в веб-мониторинге и HTTP элементах данных.

На этой странице описывается пример настройки Kerberos для выполнения Zabbix сервером веб-мониторинга страницы `www.example.com` с использованием принципала Kerberos для процесса Zabbix в Debian/Ubuntu.

Конфигурация

1. Установите KDC и клиентские утилиты:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Во время настройки пакетов ответьте на запросы, например:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. Сопоставьте удобное имя узла сети (необязательно, для локального тестирования).

Отредактируйте /etc/hosts и добавьте записи для контроллера домена и веб-сервера, если у вас нет DNS:

```
sudo vi /etc/hosts
```

Пример добавляемой строки:

```
192.168.1.100  dc01.example.com dc01
```

3. Настройте клиент Kerberos и область KDC:

```
sudo vi /etc/krb5.conf
```

Пример настроек:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

Если вы планируете использовать `.localdomain` или другие непубличные имена, добавьте явные сопоставления доменов с областями, чтобы сопоставление имени узла сети с областью работало корректно. Несоответствия в этой конфигурации приводят к ошибкам `Server not found in Kerberos database`.

4. Инициализируйте базу данных Kerberos (однократно, на узле сети KDC). При появлении запроса задайте надежный пароль главного ключа:

```
sudo krb5_newrealm
```

5. Создайте субъект `HTTP/host.fqdn@REALM`, используя точное имя узла сети, которое будут применять клиенты; предпочтительно использовать нижний регистр (например, `HTTP/web.example.com@EXAMPLE.COM`). Несоответствие регистра или имени приводит к ошибке `Server not found in Kerberos database`.

```
sudo kadmin.local
```

В `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM     #
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Переместите файл keytab на веб-узел сети (или оставьте его локально, если используется тот же компьютер) и задайте разрешения, позволяющие Apache использовать этот файл:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
####
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Установите и включите модуль Apache GSSAPI:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Не все версии mod_auth_gssapi поддерживают каждую директиву `Gssapi*`. Если Apache завершается с ошибкой `Invalid command 'GssapiCredStore'`, удалите неподдерживаемую директиву или обновите модуль.

7. Настройте VirtualHost (измените `DocumentRoot` и путь к веб-интерфейсу Zabbix при необходимости):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

В `zabbix.conf`:

```
<VirtualHost *:80>
    ServerName dc01.example.com
```

```
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Перезапустите Apache:

```
sudo systemctl restart apache2
```

8. Включите и запустите службы KDC, а также проверьте прослушиваемые порты (на узле сети KDC):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    #   : sudo netstat -tnlp | grep :80
```

9. Получите TGT для тестирования (выполняйте от имени пользователя, который будет использовать билет).

В списке билетов должна отображаться запись `krbtgt/EXAMPLE.COM@EXAMPLE.COM`. Запускайте `kinit` от имени того же пользователя ОС, которому нужен билет (например, `zabbix` для веб-проверок или `www-data`/Apache для тестирования интерактивного SSO в браузере). Билеты, выданные другому пользователю ОС, не будут видны, если не настроить `KRB5CCNAME` и разрешения.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Протестируйте обмен SPNEGO с помощью curl (с клиента с действительным TGT). Ответ `200 OK` (или перенаправление в приложение) означает, что SPNEGO успешно выполнен:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. При необходимости, если веб-интерфейс Zabbix должен принимать входящие подключения с HTTP-аутентификацией, включите HTTP-аутентификацию в веб-интерфейсе Zabbix.

В файле конфигурации веб-интерфейса (`zabbix.conf.php`) задайте:

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Затем настройте HTTP-аутентификацию в веб-интерфейсе.

12. Настройте браузер (в качестве примера используется Firefox): задайте для `network.negotiate-auth.trusted-uris` узел или узлы сети, выполняющие Negotiate (`dc01.example.com`), чтобы браузер автоматически отправлял токены Kerberos.

В `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Теперь при переходе на `http://dc01.example.com` должен выполняться непосредственный вход в Zabbix без отображения формы.

13. Поддерживайте ключи и билеты в актуальном состоянии. Срок действия билета Kerberos по умолчанию составляет примерно 10 часов. Добавьте задание cron или таймер systemd, чтобы избежать истечения срока действия:

```
####    -
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Проверки обслуживания:

- `klist -k /etc/apache2/http.keytab` - проверьте наличие субъекта службы в keytab.

- `sudo tail -f /var/log/apache2/error.log` - отслеживайте ошибки GSSAPI (`gss_acquire_cred[_from]() failed to get server creds` означает проблему с keytab или разрешениями либо отсутствие субъекта).
- Если `curl --negotiate` возвращает 401/403, это часто означает неправильный субъект, отсутствие билета, несоответствие заголовка узла сети или проблему с разрешениями файловой системы; проверьте журналы и сопоставления доменов в `/etc/krb5.conf`.

Примечания по безопасности и разрешениям файлов

Файлы keytab должны быть доступны для чтения только учетной записи, которой они необходимы. Примеры разрешений: 0400 с владельцем `zabbix:zabbix` для keytab пользователя zabbix или 0440 с владельцем `root:www-data` для keytab Apache.

Не храните на узле сети пароли в открытом виде с длительным сроком действия. По возможности используйте keytab или учетные записи компьютеров, присоединенных к домену.

При выполнении тестов или скриптов, которые задают `KRB5CCNAME` или копируют keytab, после завершения операции еще раз проверьте владельца и разрешения — отказ веб-сервера в принятии учетных данных часто связан с проблемами разрешений файлов.

Параметры modbus.get

Обзор

В таблице ниже представлены детальные описания параметров элемента данных `modbus.get`.

Parameters

Параметр	Описание	По умолчанию	Пример
точка входа	Протокол и адрес точки входа, определяемый в виде :// _ Возможные значения протокола: *rtu*, *ascii* (только Агент 2), *tcp* Формат строки подключения: для *tcp* — : для последовательной линии: *rtu*, *ascii* — _ : : где «скорость» — 1200, 9600 и т.д. «параметры» — количество бит (5,6,7 или 8), чётность (n,e или o для нет(none)/чёт(even)/нечет(odd)), стоп-битов (1 или 2)	протокол: нет протокол *rtu/ascii*: имя_порта: нет скорость: 115200 параметры: 8n1 протокол *tcp*: адрес: нет порт: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
id ведомого	Адрес modbus устройства, для которого он предназначен (от 1 до 247), см. MODBUS Messaging Implementation Guide [en] (страница 23) устройство tcp (не GW) будет игнорировать это поле	для последовательной линии: 1 для tcp: 255 (0xFF)	2
функция	Пусто или значение поддерживаемой функции: 1 — Чтение Флага, 2 — Чтение Дискретного Входа, 3 — Чтение Регистров Временного Хранения, 4 — Чтение Входных Регистров	пусто	3

Параметр	Описание	По умолчанию	Пример
адрес	адрес первого регистра, флага или входа. Если параметр «функция» пустой, то «адрес» должен быть в диапазоне для: Флага — 00001 — 09999 Дискретного входа — 10001 — 19999 Входного регистра — 30001 — 39999 Регистра Временного Хранения — 40001 — 49999 Если параметр «функция» не пуст, то поле «адрес» будет от 0 до 65535 использоваться без модификации (PDU)	пустая функция: 00001 непустая функция: 0	9999
количество	Количество последовательных «типов», которые будут считаны с устройства, где: для флага или дискретного входа «тип» = 1 бит в остальных случаях: (количество*sizeof(тип))/2 = реальное количество регистров для чтения Если «сдвиг» не равен 0, то значение будет добавлено к «реальному количеству» Допустимый диапазон для «реального количества» составляет 1:65535	1	2
тип	Тип данных: для Чтения Флага и Чтения Дискретного Входа — *bit* для Чтения Регистров Временного Хранения и Чтения Входных Регистров: *int8* — 8 бит *uint8* — 8 бит (без знака) *int16* — 16 бит *uint16* — 16 бит (без знака) *int32* — 32 бита *uint32* — 32 бита (без знака) *float* — 32 бита *uint64* — 64 бита (без знака) *double* — 64 бита	bit uint16	uint64
порядок следования байтов	Тип порядка следования байтов: *be* — Big Endian *le* — Little Endian *mbe* — Mid-Big Endian *mle* — Mid-Little Endian Ограничения: для 1 бит — be для 8 бит — be,le для 16 бит — be,le	be	le
сдвиг	Количество регистров, начиная с «адреса», результаты которых будут отброшены. Размер каждого регистра составляет 16 бит (требуется для поддержки оборудования, не поддерживающего произвольный доступ для чтения).	0	4

Создание пользовательских имен счетчиков производительности для VMware

Обзор

Путь счетчика производительности VMware имеет формат `group/counter[rollup]`, где:

- `group` - группа счетчиков производительности, например, *cpu*
- `counter` - имя счетчика производительности, например, *usagemhz*
- `rollup` - тип агрегации счетчика производительности, например, *average*

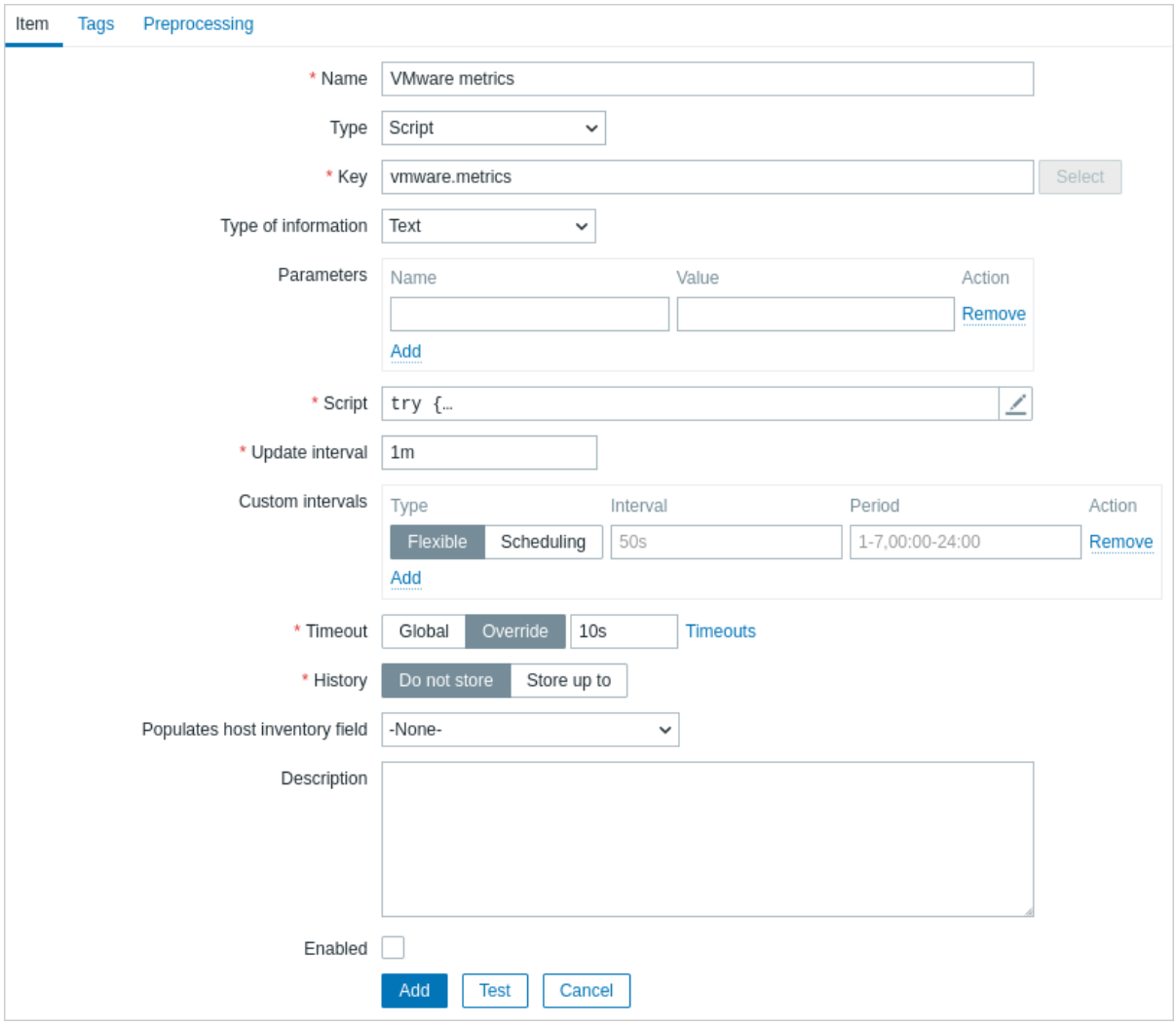
Таким образом, приведенный выше пример даст следующий путь счетчика: `cpu/usagemhz[average]`

Описания групп счетчиков производительности, имена счетчиков и типы агрегации можно найти в документации VMware.

Можно получить внутренние имена и создать пользовательские имена счетчиков производительности, используя элемент данных script в Zabbix.

Конфигурация

1. Создайте отключенный элемент данных типа Script на основном узле сети VMware (где присутствует элемент данных **eventlog[]**) со следующими параметрами:



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: скопируйте и вставьте скрипт, приведенный ниже
- *Timeout*: 10
- *History*: Do not store
- *Enabled*: unmarked

Скрипт

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');
```

```
    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

После настройки элемента данных нажмите кнопку *Test*, затем *Get value*.



Скопируйте полученный XML в любой XML-форматтер и найдите нужную метрику.

Пример XML для одной метрики:

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
    <key>6</key>
    <nameInfo>
        <label>Usage in MHz</label>
        <summary>CPU usage in megahertz during the interval</summary>
        <key>usagemhz</key>
    </nameInfo>
    <groupInfo>
        <label>CPU</label>
        <summary>CPU</summary>
        <key>cpu</key>
    </groupInfo>
    <unitInfo>
        <label>MHz</label>
        <summary>Megahertz</summary>
        <key>megaHertz</key>
    </unitInfo>
    <rollupType>average</rollupType>
    <statsType>rate</statsType>
    <level>1</level>
    <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

Используйте XPath, чтобы извлечь путь счетчика из полученного XML. Для приведенного выше примера XPath будет следующим:

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Итоговый путь счетчика производительности в этом случае: `cpu/usagemhz[average]`

Возвращаемые значения для system.sw.packages.get

Обзор

В этом разделе приведены сведения о значениях, возвращаемых для элемента данных Zabbix агента `system.sw.packages.get`.

Подробности

Вывод этого элемента данных представляет собой массив объектов, каждый из которых содержит следующие свойства:

- **name** - имя пакета
- **manager** - менеджер пакетов, который сообщил эти данные (`rpm`, `dpkg`, `pacman`, `pkgtool` или `portage`)
- **version** - версия пакета
- **size** - размер распакованного пакета в байтах (если недоступно, установите в 0)
- **arch** - архитектура пакета
- **buildtime** - объект с 2 записями:
  - **timestamp** - UNIX timestamp, когда пакет был собран (если недоступно, установите в 0)
  - **value** - человекочитаемые дата и время, когда пакет был собран (если недоступно, установите в пустую строку)

- **installtime** - объект с 2 записями:
  - **timestamp** - UNIX timestamp, когда пакет был установлен (если недоступно, установите в 0)
  - **value** - человекочитаемые дата и время, когда пакет был установлен (если недоступно, установите в пустую строку)

Пример:

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

Возвращаемые значения для net.dns.get

Overview

В этом разделе приведены сведения о значениях, возвращаемых для элемента данных Zabbix агента 2 `net.dns.get`.

Подробности

Вывод этого элемента данных — объект, содержащий информацию о записи DNS на основе параметров, указанных в ключе элемента данных.

Например, элемент данных `net.dns.get[,example.com]` может вернуть следующий JSON отклонённого запроса:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
```

```
    "zbx_error_code": 0
}
```

С указанием IP-адреса DNS-сервера, элемент данных `net.dns.get[192.0.2.1,example.com]` может вернуть следующий JSON:

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
                "ns": "ns.example.org.",
                "refresh": 7200,
                "retry": 3600,
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```

Если возникла проблема с подключением, элемент данных `net.dns.get[192.0.2.1,example.com]` может содержать ошибку:

```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```

Возможны следующие типы кодов ошибок:

Сценарий	"zbx_error_code"	"zbx_error_msg"
Ошибок нет, и ответ DNS получен и проанализирован.	0	
DNS не работает.	-1	"Communication error"
Ошибка при анализе JSON	-2	"Received unexpected response"

С дополнительными параметрами элемент данных `net.dns.get[192.0.2.1,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnss` может вернуть следующий JSON:

```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
```

```
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
            ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "a": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Смотрите также

Более подробную информацию о записях DNS смотрите [en]:

- Доменные имена — Реализация и спецификация
- Параметры системы доменных имен (DNS)

Возвращаемые значения для net.if.get

Обзор

В этом разделе приведены сведения о возвращаемом значении для элемента данных `net.if.get`. Этот элемент данных поддерживается в Linux агентом Zabbix и Zabbix agent 2.

Подробности

Этот элемент данных возвращает подробную информацию о сетевых интерфейсах, имена которых соответствуют заданному регулярному выражению. Вывод структурирован как JSON-объект, содержащий массивы `config` и `values`.

Вывод содержит следующие свойства:

- **name** - имя интерфейса, считывается из `/proc/net/dev`;
- **ifalias** - псевдоним интерфейса, считывается из `/sys/class/net/<name>/ifalias`;
- **mac** - MAC-адрес интерфейса, считывается из `/sys/class/net/<name>/address`;
- **type** - тип интерфейса (`physical`, `virtual`, `loopback`);
- **speed** - скорость соединения в Мбит/с, специфично для Ethernet, считывается из `/sys/class/net/<name>/speed`;
- **duplex** - последнее или текущее значение дуплекса интерфейса (`half`, `full`), специфично для Ethernet, считывается из `/sys/class/net/<name>/duplex`;
- **administrative_state** - настроенное состояние (`down`, `up`), считывается из `/sys/class/net/<name>/flags`;
- **operational_state** - рабочее состояние интерфейса согласно RFC2863 (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), считывается из `sys/class/net/<name>/operstate`;
- **carrier** - наличие физического соединения (0 - физическое соединение неактивно, 1 - физическое соединение активно), считывается из `/sys/class/net/<name>/carrier`;
- **carrier_changes** - общее количество изменений состояния соединения с up на down и наоборот, считывается из `/sys/class/net/<name>/carrier_changes`;
- **carrier_up_count** - количество раз, когда соединение становилось активным, считывается из `/sys/class/net/<name>/carri`
- **carrier_down_count** - количество раз, когда соединение становилось неактивным, считывается из `/sys/class/net/<name>/`
- **in** - входящий трафик, считывается из `/proc/net/dev`. Содержит счетчики интерфейса:
  - **bytes** - количество корректных байтов;
  - **packets** - количество корректных пакетов;
  - **errors** - общее количество некорректных пакетов;
  - **dropped** - количество полученных, но не обработанных пакетов + количество пакетов, пропущенных узлом сети;
  - **overruns** - количество событий переполнения;
  - **frame** - количество пакетов с ошибками кадрирования;
  - **compressed** - количество корректно полученных сжатых пакетов;
  - **multicast** - количество полученных multicast-пакетов.
- **out** - исходящий трафик, считывается из `/proc/net/dev`. Содержит счетчики интерфейса:
  - **bytes** - количество корректных байтов;
  - **packets** - количество корректных пакетов;
  - **errors** - общее количество некорректных пакетов;
  - **dropped** - количество полученных, но не обработанных пакетов + количество пакетов, пропущенных узлом сети;
  - **overruns** - количество событий переполнения;
  - **collisions** - количество коллизий при передаче пакетов;
  - **carrier** - количество ошибок передачи кадров из-за потери несущей во время передачи;
  - **compressed** - количество переданных сжатых пакетов.

Недоступные метрики исключаются из JSON-вывода. В следующем примере для интерфейса "lo" не настроен псевдоним, поэтому свойство `ifalias` исключено.

Пример:

```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
```

```
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
```

```
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


См. также

Дополнительную информацию о sysfs ABI для сетевых устройств см. в:

- Документации ядра Linux

Возвращаемые значения для vfs.dev.get

Обзор

В этом разделе приведены сведения о возвращаемом значении для элемента данных `vfs.dev.get`. Элемент данных поддерживается в Linux агентом Zabbix и Zabbix agent 2, а в AIX — агентом Zabbix.

Подробности

Этот элемент данных возвращает подробную информацию о блочных устройствах, имена которых соответствуют заданному регулярному выражению.

Вывод структурирован как объект JSON, содержащий массивы `config` и, в зависимости от параметра `<mode>`, `values`.

Массив `config` содержит свойства, доступные для выбранного режима. Массив `values` возвращается только для режимов, которые предоставляют статистику по блочным устройствам.

Недоступные метрики исключаются из вывода JSON.

В следующих разделах описаны свойства, возвращаемые для каждого режима.

режим дисков

`vfs.dev.get[disks,.*]` возвращает данные конфигурации блочных устройств.

Свойства:

- **name** - имя блочного устройства;
- **devid** - идентификатор блочного устройства, например идентификатор устройства на основе WWN или сгенерированный ядром;
- **type** - тип блочного устройства (`disk`, `partition`, `rom`);
- **path** - путь к узлу блочного устройства;
- **model** - модель устройства;
- **serial** - серийный номер устройства;
- **wwn** - World Wide Name (WWN) устройства;
- **size_bytes** - размер устройства в байтах;
- **logical_block_size** - размер логического блока в байтах;
- **physical_block_size** - размер физического блока в байтах.

Пример для Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
```

```
    }
  ]
}
```

Пример для AIX:

```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```

режим устройств

`vfs.dev.get[devices,.*]` возвращает данные конфигурации блочных устройств, включая разделы.

Свойства:

- **name** - имя блочного устройства.
- **devid** - идентификатор блочного устройства.
- **type** - тип блочного устройства (`disk`, `partition`, `rom`).
- **partitions** - разделы на устройстве. Содержит имена разделов и их размер в байтах.

Пример для Linux:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
```

```
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```

режим disk_stats

`vfs.dev.get[disk_stats,.*]` возвращает данные конфигурации блочных устройств и статистику дисков.

Свойства:

- **name** - имя блочного устройства.
- **devid** - идентификатор блочного устройства.
- **type** - тип блочного устройства (`disk`, `partition`, `rom`).
- **size_bytes** - размер устройства в байтах.
- **logical_block_size** - размер логического блока в байтах.
- **stats** - статистика ввода-вывода блочного устройства. Содержит:
  - **reads_completed** - количество завершенных операций чтения;
  - **writes_completed** - количество завершенных операций записи;
  - **bytes_read** - количество прочитанных байтов;
  - **bytes_written** - количество записанных байтов;
  - **io_time_ms** - общее время, затраченное на операции ввода-вывода, в миллисекундах.

Пример для Linux

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
```

```
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    }
  ]
}
```


Пример для AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
```

```
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
  ]
}
```

режим device_stats

`vfs.dev.get[device_stats,.*]` возвращает данные конфигурации блочного устройства и статистику устройства.

Свойства:

- **name** - имя блочного устройства.
- **devid** - идентификатор блочного устройства.
- **type** - тип блочного устройства (`disk`, `partition`, `rom`).
- **size_bytes** - размер устройства в байтах.
- **stats** - статистика I/O блочного устройства. Содержит:
  - **reads_completed** - количество завершенных операций чтения;
  - **writes_completed** - количество завершенных операций записи;
  - **bytes_read** - количество прочитанных байт;
  - **bytes_written** - количество записанных байт;
  - **io_time_ms** - общее время, затраченное на I/O, в миллисекундах.

Пример для Linux:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
```

```
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
```

```
      "io_time_ms": 1256
    }
  },
  {
    "name": "sda1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sda2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb",
    "stats": {
      "reads_completed": 51017,
      "writes_completed": 10682,
      "bytes_read": 10514627,
      "bytes_written": 1530472,
      "io_time_ms": 284988
    }
  },
  {
    "name": "sdb1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb3",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
```

```
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    },
    {
      "name": "nvme0n1p1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```


Примечания к элементам данных system.cpu.util в Windows

Элемент данных system.cpu.util предоставляет процент загрузки CPU.

Когда на агенте Zabbix для Windows запускается процесс сборщика, инициализируется буфер для N элементов данных для N логических процессоров (потоков). Значения периодически обновляются в буфере с использованием счетчиков производительности Windows. Эти значения возвращаются при запросе `system.cpu.util[n]`, где 0 <= n < N.

Еще один элемент в буфере используется для `system.cpu.util[total]`.

Zabbix автоматически выбирает разные счетчики производительности, чтобы поддерживать как системы NUMA, так и более старые версии Windows без поддержки групп процессоров.

Если количество логических процессоров (потоков) меньше или равно 64, а число групп процессоров равно 1, то:

```
N —                        (    ).

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

Иначе:

```
G —                    ,   N —                    (    )      .

\Processor Information(_Total)\% Processor Time
\Processor Information(0,0)\% Processor Time
\Processor Information(0,1)\% Processor Time
\Processor Information(0,2)\% Processor Time
...
\Processor Information(0,N-1)\% Processor Time
...
\Processor Information(G-1,0)\% Processor Time
\Processor Information(G-1,1)\% Processor Time
\Processor Information(G-1,2)\% Processor Time
...
\Processor Information(G-1,N-1)\% Processor Time
```

Поддержка больших значений JSON

Чтобы обеспечить надежную обработку больших значений с типом данных JSON data type (1 МиБ или более), при необходимости проверьте и примените следующие изменения конфигурации:

1. Если вы используете MySQL или MariaDB, увеличьте максимальный размер разрешенного пакета в `/etc/mysql/my.cnf` (на некоторых системах по умолчанию он может быть 1 МБ):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. При необходимости настройте дополнительные server system variables MySQL/MariaDB и InnoDB system variables. Например:

```
innodb_io_capacity=1900
connect_timeout=600
wait_timeout=57600
interactive_timeout=57600
net_read_timeout=7200
net_write_timeout=7200
net-buffer-length=32704
```

3. Увеличьте значение `HistoryCacheSize` до 2G в файле конфигурации сервера Zabbix.

4. Если вы используете Zabbix proxy, увеличьте значение `HistoryCacheSize` до 2G и `ProxyMemoryBufferSize` до 2G в файле конфигурации прокси.

5. Если вы используете Zabbix агент, увеличьте тайм-ауты обмена данными между сервером/прокси Zabbix и агентом (например, item timeouts или значение параметра `Timeout` в файлах конфигурации Zabbix server/proxy и agent). В противном случае сообщения между ними могут не завершаться, и вы можете увидеть ошибки, такие как *"message length does not match expected length"*. Также при необходимости проверьте другие тайм-ауты, например тайм-аут script timeout или item test.

Поддерживаемые функции

См. Выражение триггера и функции.

Макросы

Можно использовать встроенные поддерживаемые макросы и пользовательские макросы, поддерживаемые в зависимости от расположения.

Поддерживаемые макросы для местоположения

Обзор

Таблица макросов, поддерживаемых в зависимости от расположения, содержит справочную информацию по всем макросам (встроенным, выражениям и пользовательским), которые поддерживаются в конкретном месте.

Если поддерживаются все макросы из связанной группы, указывается только название группы. Щелкните по названию каждого макроса или группы, чтобы увидеть полные сведения о макросах.

Список подробностей о макросах содержит подробный перечень всех **встроенных** макросов, сгруппированных по области применения. Обратите внимание, что для настройки значений макросов (например, для сокращения или извлечения определенных подстрок) можно использовать функции макросов.

Макросы, поддерживаемые в зависимости от расположения

Расположение	Поддерживаемые макросы	Комментарии
Действие		
Условие периода времени, *Длительность шага операции по умолчанию*, *Длительность шага*	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются
Уведомления и команды		

Расположение	Поддерживаемые макросы	Комментарии
На основе триггера	Макросы действия Макросы даты/времени {ESC.HISTORY} Макросы события, макросы события причины/симптома Макросы обновления события: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Макрос выражения: {?EXPRESSION} {FUNCTION.VALUE} Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы целевого узла сети (только для команд) Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE}, {ITEM.STATE.ERROR}) Макросы прокси Макросы триггера: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Пользовательские макросы	См. также: Использование макросов в сообщениях
Обновление проблемы	Макросы действия Макросы даты/времени {ESC.HISTORY} Макросы события, макросы события причины/симптома, макросы события восстановления Макросы обновления события: все (кроме {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Макрос выражения: {?EXPRESSION} Макросы функций Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы целевого узла сети (только для команд) Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE}, {ITEM.STATE.ERROR}) Макросы прокси Макросы триггера: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} {USER.FULLNAME} Пользовательские макросы	
Восстановление проблемы	Макросы события восстановления Макросы события {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	

Расположение	Поддерживаемые макросы	Комментарии
Обнаружение	Макросы действия Макросы даты/времени Макросы обнаружения Макросы события: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Макросы целевого узла сети (только для команд) Макросы прокси	
Автоматическая регистрация	Макросы действия Макросы даты/времени Макросы события: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Макросы узла сети: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Макросы целевого узла сети (только для команд) Макросы прокси	
На основе сервиса	Макросы действия Макросы даты/времени {ESC.HISTORY} Макросы события: все, кроме {EVENT.ACK.STATUS}, {EVENT.OPDATA} Макросы сервиса Пользовательские макросы	
Восстановление сервиса	Макросы события (кроме {EVENT.ACK.STATUS}, {EVENT.OPDATA}), макросы события восстановления	
Обновление сервиса	Макросы действия Макросы даты/времени {ESC.HISTORY} Макросы события (кроме {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Макросы обновления события: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Макросы сервиса Пользовательские макросы	
Внутренние уведомления	Макросы действия Макросы даты/времени {ESC.HISTORY} Макросы события (кроме {EVENT.NSEVERITY}, {EVENT.SEVERITY}), макросы события восстановления Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Макросы элемента данных: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Макросы прокси	
На основе элемента данных	Макросы элемента данных: {ITEM.STATE}, {ITEM.STATE.ERROR}	
На основе правила LLD	Макросы уведомления низкоуровневого обнаружения	

Расположение	Поддерживаемые макросы	Комментарии
На основе триггера	Макросы триггера: {TRIGGER.DESCRIPTION}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATE}, {TRIGGER.STATE.ERROR}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME} Пользовательские макросы	
Параметры скрипта оповещения **Коннектор**	Макросы оповещения Пользовательские макросы (только глобальные)	
Поле URL, Username, Password, Bearer token, HTTP proxy, SSL certificate, SSL key file, SSL key password	Пользовательские макросы (только глобальные)	
Поле Timeout	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Имя графика	Макрос выражения: {?EXPRESSION}	В этом макросе в данном расположении поддерживается только одна функция **avg**, **last**, **max** или **min** с параметром в секундах (можно использовать суффиксы времени). В качестве узла сети в макросе можно использовать макрос {HOST.HOST<1-9>} (см. также индексированные макросы). Пример использования: {?avg(/{HOST.HOST}/item.key,1h)} Неподдерживаемое использование: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Узел сети, прототип узла сети		
IP/DNS интерфейса	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Пользовательские макросы [1]	[1] Для IP-адреса один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются

Расположение	Поддерживаемые макросы	Комментарии
Порт интерфейса	Пользовательские макросы	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
SNMP community *SNMPv3 Context name, Security name, Authentication passphrase, Privacy passphrase* *IPMI username, password*	Пользовательские макросы	
Элемент данных, прототип элемента данных, правило LLD		
Имя	Пользовательские макросы	Не поддерживается в именах правил LLD.
Описание	Пользовательские макросы	
Параметры ключа	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	Поддерживаемые макросы {HOST.*} будут разрешаться в интерфейс, выбранный для элемента данных. При использовании в элементах данных без интерфейсов они будут разрешаться либо в интерфейс агента Zabbix, SNMP, JMX или IPMI узла сети в порядке приоритета, либо в 'UNKNOWN', если у узла сети нет ни одного интерфейса. Макросы не поддерживаются в ключе элемента данных типа Zabbix trapper.
Поля Интервал обновления, Пользовательские интервалы, Таймаут (доступно для поддерживаемых типов элементов данных), Хранить до (для истории/трендов)	Пользовательские макросы	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.

Расположение	Поддерживаемые макросы	Комментарии
Поля Удалять потерянные ресурсы, Отключать потерянные ресурсы	Пользовательские макросы	Эти поля поддерживаются только для правил LLD. Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Параметры шага предварительной обработки (включая пользовательские скрипты), *параметры обработки пользовательских ошибок* (поля *Установить значение* и *Установить ошибку в*)	Пользовательские макросы	
Регулярные выражения фильтра	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	Эти поля поддерживаются только для правил LLD.
Переопределить регулярные выражения фильтра	Пользовательские макросы	Эти поля поддерживаются только для правил LLD.
Переопределить операции: *Интервал обновления*, *Период хранения истории*, *Период хранения трендов* (для прототипов элементов данных)	Пользовательские макросы	Эти поля поддерживаются только для правил LLD. Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Типы элементов данных		
Элемент Browser *имена и значения параметров*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Макросы элемента данных: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Пользовательские макросы	
Элемент Script *имена и значения параметров*		

Расположение	Поддерживаемые макросы	Комментарии
Формула **Calculated/aggregate** элемента (*константы выражения*, *параметры функции*, *параметры ключа элемента данных*, (только для aggregate-элемента) *условия фильтра* (имя группы узлов сети и имя тега))	Пользовательские макросы	
Элемент Database monitor *поля Username, Password, SQL query*	Пользовательские макросы	
Элемент HTTP agent *поля URL, Query fields, Request body, Headers (names and values), SSL certificate file, SSL key file*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Пользовательские макросы	URL, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешен как ”******”
Элемент HTTP agent *поле HTTP Proxy*	Макросы элемента данных: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Пользовательские макросы	
Элемент HTTP agent *поля Required status codes, HTTP authentication username/password, SSL key password*	Пользовательские макросы	
Элемент HTTP agent *поле Allowed hosts*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Пользовательские макросы	
Элемент JMX agent *поле Endpoint*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Пользовательские макросы	
Элемент Trapper *поле Allowed hosts*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} Пользовательские макросы	
Элемент SNMP agent *поле SNMP OID*	Пользовательские макросы	
Элемент SSH and Telnet *script*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
Элемент SSH agent *поля Username, Public key file, Private key file, Password*	Пользовательские макросы	
Элемент Telnet agent *поля Username, Password*	Пользовательские макросы	
Карта		

Расположение	Поддерживаемые макросы	Комментарии
URL, URL name	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Макросы инвентаря узла сети Макросы карты {TRIGGER.ID}	
Метка элемента	Макрос выражения: {?EXPRESSION} [1] Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Макросы инвентаря узла сети Макросы карты Макросы триггера: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] В этом макросе в данном расположении поддерживается только одна функция **avg**, **last**, **max** или **min** с параметром в секундах (можно использовать суффиксы времени). В качестве узла сети в макросе можно использовать макрос {HOST.HOST<1-9>}. Пример использования: {?avg(/{HOST.HOST}/item.key,1h)} Неподдерживаемое использование: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Метка ссылки	Макрос выражения: {?EXPRESSION} [1]	
Метка фигуры	Макрос выражения: {?EXPRESSION} [1]	
Текстовое поле в фигурах карты	{MAP.NAME}	
Сетевое обнаружение		
Поле Интервал обновления	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Поля SNMP community, SNMP OID *Поля SNMPv3 Context name, Security name, Authentication passphrase, Privacy passphrase*	Пользовательские макросы (только глобальные)	
Прокси		
Address for active agents > *Port* (если прокси принадлежит группе) Для пассивного режима прокси: поля интерфейса *Address* и *Port* Поля *Timeouts for item types* в режиме переопределения	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Группа прокси		

Расположение	Поддерживаемые макросы	Комментарии
Поля Failover period, Minimum number of proxies	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Скрипты		
Поля Commands, confirmation text (Тип: script, manual **host action**)	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети [2] {MANUALINPUT} Макросы имени пользователя Пользовательские макросы	[1] Только текст подтверждения [2] Поддерживается для Zabbix server и Zabbix proxy
Поля Commands (Тип: script, manual **event action**)	Макросы даты/времени Макросы события, макросы события причины/симптома, макросы события восстановления (если восстановление произошло) Макросы обновления события: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Макросы прокси Макросы триггера: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Макросы имени пользователя Пользовательские макросы	
Текст подтверждения (Тип: script, manual **event action**)	Макросы события: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети {MANUALINPUT} Макросы имени пользователя Пользовательские макросы	
Поля URL, confirmation text (Тип: URL, manual **host/event action**)		

Расположение	Поддерживаемые макросы	Комментарии
Имя тега и значение	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Макросы триггера: {TRIGGER.ID} (только значение тега триггера) Пользовательские макросы	Макросы имени и значения тега разрешаются только в процессе генерации события.
Триггер		
Имя	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Позиционные макросы/ссылки: $1...$9 Пользовательские макросы	
Имя события	Макросы даты/времени: {TIME}, {TIMESTAMP} Макрос выражения: {?EXPRESSION} Макросы функций Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Макросы триггера: {TRIGGER.EXPRESSION.EXPLAIN}	
Оперативные данные	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Пользовательские макросы	
Выражение	{TRIGGER.VALUE} Пользовательские макросы [1]	[1] Только в константах и параметрах функций; секретные макросы не поддерживаются
Поля Tag for matching, Menu entry name, Menu entry URL	Пользовательские макросы	URL, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешен как "******".
Описание	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Пользовательские макросы	
URL	{EVENT.ID} Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Команды пользовательского параметра	Позиционные макросы/ссылки: $1...$9	
Веб-сценарий		
Поля Name, Agent, HTTP proxy	Пользовательские макросы	

Расположение	Поддерживаемые макросы	Комментарии
Поле Update interval	Пользовательские макросы	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Поля Variables, Headers, SSL certificate, SSL key file	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
Поля User, Password, SSL key password	Пользовательские макросы	
Шаг веб-сценария *поля Name, Variables (только значения)*	Пользовательские макросы	
Шаг веб-сценария *поля Timeout, Required status codes*	Пользовательские макросы	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Шаг веб-сценария *поля URL, Headers (имена и значения), Required string*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
Шаг веб-сценария *поле Post*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Тип медиа: вебхук		
Значения параметров	Макросы оповещения Все макросы, поддерживаемые в уведомлениях о проблемах на основе триггера (см. выше)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Скрипт JavaScript, имя параметра *Скрипт* JavaScript	Пользовательские макросы (только глобальные)	
Поля Menu entry name, Menu entry URL	{EVENT.TAGS.<tag name>}	
Виджеты		
Виджет Gauge *поле Description*	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE.ERROR}) Пользовательские макросы (только глобальные)	
Виджет Honeycomb *поле Primary/secondary label*		
Виджет Item value *поле Description*		
Виджет Top hosts *столбец Text data*	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентаря узла сети Пользовательские макросы (только глобальные)	
Виджет URL (Dynamic) *поле URL*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	

Расположение	Поддерживаемые макросы	Комментарии
Столбец **Cus-** **tom** **text***		
Мониторинг > Проблемы	Макросы узла сети (кроме {HOST.METADATA}) Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE.ERROR}) Макросы триггера: {TRIGGER.ID} Макросы события: {EVENT.ID} Пользовательские макросы	
Мониторинг > Узлы сети	Макросы узла сети (кроме {HOST.METADATA}) Макросы инвентаря узла сети Пользовательские макросы	
Мониторинг > Последние данные	Макросы узла сети (кроме {HOST.METADATA}) Макросы инвентаря узла сети Макросы элемента данных (кроме {ITEM.STATE.ERROR}) Пользовательские макросы	
Сбор данных > Шаблоны	Макросы узла сети: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME}	
Сбор данных > Узлы сети	Макросы узла сети (кроме {HOST.METADATA}) Макросы инвентаря узла сети Пользовательские макросы	
Другие расположения		
Поле Working time	Пользовательские макросы (только глобальные)	Один макрос должен заполнять все поле; несколько макросов или сочетание с текстом не поддерживаются.
Поле Timeouts for item types (глобальная настройка)		
Медиа пользователя *поле When active*	Пользовательские макросы (только глобальные)	
Тип медиа Email *поля Username, Password*	Пользовательские макросы (только глобальные)	Рекомендуются секретные макросы.
Запланированные отчеты *поля Subject, Message*	Макросы даты/времени: {TIME}, {TIMESTAMP}	

Подробности о макросах

Этот список содержит подробную информацию о каждом встроенном макросе, сгруппированном по области применения:

- Макросы действий
- Макросы оповещений
- Макросы даты/времени
- Макросы обнаружения
- Макросы событий
  - Макросы обновления событий
  - Макросы событий причины/симптома
  - Макросы событий восстановления
- Макросы эскалации
- Макросы функций
- Макросы узла сети
  - Макросы целевого узла сети
- Макросы группы узлов сети
- Макросы инвентаризации узла сети
- Макросы элементов данных

- Макросы уведомлений низкоуровневого обнаружения
- Макросы карт
- Макросы прокси
- Макросы скриптов
- Макросы сервисов
- Макросы триггеров
- Макросы имени пользователя

Макросы действий

{ACTION.ID}

Числовой ID сработавшего действия.

{ACTION.NAME}

Имя сработавшего действия.

Макросы оповещений

{ALERT.MESSAGE}

Значение *Сообщение по умолчанию* из конфигурации действия.

{ALERT.SENDTO}

Значение *Send to* из конфигурации медиа пользователя.

{ALERT.SUBJECT}

Значение *Default subject* из конфигурации действия.

Макросы даты и времени

{DATE}

Текущая дата в формате `yyyy.mm.dd`.

{TIME}

Текущее время в формате `hh:mm:ss`.

{TIMESTAMP}

Текущее время в формате UNIX timestamp.

Макросы обнаружения

{DISCOVERY.DEVICE.IPADDRESS}

IP-адрес обнаруженного устройства.
 Доступен всегда, не зависит от того, был ли добавлен узел сети.

{DISCOVERY.DEVICE.DNS}

DNS-имя обнаруженного устройства.
 Доступно всегда, не зависит от того, был ли добавлен узел сети.

{DISCOVERY.DEVICE.STATUS}

Статус обнаруженного устройства (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

Время, прошедшее с момента последнего изменения статуса обнаружения для конкретного устройства.
 Например: 1h 29m 01s.
 Для устройств со статусом DOWN это период их недоступности.

{DISCOVERY.RULE.NAME}

Имя правила обнаружения, которое обнаружило наличие или отсутствие устройства/сервиса.

{DISCOVERY.SERVICE.NAME}

Имя обнаруженного сервиса.
 Например: HTTP.

{DISCOVERY.SERVICE.PORT}

Порт обнаруженного сервиса.
 Например: 80.

{DISCOVERY.SERVICE.STATUS}

Статус обнаруженного сервиса (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

Время, прошедшее с момента последнего изменения статуса обнаружения для определённого сервиса.
 Например: 1h 29m 01s.
 Для сервисов со статусом DOWN это период их недоступности.

Макросы эскалации

{ESC.HISTORY}

Журнал ранее отправленных уведомлений, их шаг эскалации и статус (*отправлено*, *в процессе* или *не удалось*).

Макросы событий

{EVENT.ACK.STATUS}

Статус подтверждения события (Да/Нет).

{EVENT.AGE}

Возраст события, которое запустило действие, с точностью до секунды.
 Полезно в сообщениях эскалации. Например: 9m 13s.

{EVENT.DATE}

Дата события, которое запустило действие, в формате `yyyy.mm.dd`.
 Например: 2025.04.14.

{EVENT.DURATION}

Длительность события (разница во времени между событиями проблемы и восстановления) с точностью до секунды.
 Полезно в сообщениях о восстановлении проблемы.

{EVENT.ID}

Числовой ID события, которое запустило действие.

{EVENT.NAME}

Имя события, которое запустило действие.

{EVENT.NSEVERITY}

Числовое значение важности события.
 Возможные значения: *0* — Не классифицировано, *1* — Информация, *2* — Предупреждение, *3* — Средняя, *4* — Высокая, *5* — Катастрофа.

{EVENT.OBJECT}

Числовое значение объекта события.
 Возможные значения: *0* — триггер, *1* — обнаруженный узел сети, *2* — обнаруженный сервис, *3* — авторегистрация, *4* — элемент данных, *5* — правило обнаружения низкого уровня.

{EVENT.OPDATA}

Оперативные данные базового триггера проблемы.

{EVENT.SEVERITY}

Название важности события.

{EVENT.SOURCE}

Числовое значение источника события.
 Возможные значения: *0* — триггер, *1* — обнаружение, *2* — авторегистрация, *3* — внутреннее событие, *4* — сервис.

{EVENT.STATUS}

Текстовое значение события, которое запустило действие.

{EVENT.TAGS}

Разделённый запятыми список тегов события.
Разворачивается в пустую строку, если теги отсутствуют.

{EVENT.TAGSJSON}

Массив JSON, содержащий объекты тегов события.
Разворачивается в пустой массив, если теги отсутствуют.

{EVENT.TAGS.<tag name>}

Значение тега события, на которое ссылается имя тега.
 Имя тега, содержащее неалфавитно-цифровые символы (включая неанглийские многобайтовые UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные слэши внутри заключенного в кавычки имени тега должны быть экранированы обратным слэшем.

{EVENT.TIME}

Время события, которое запустило действие, в формате `hh:mm:ss`.
 Например: 12:57:53.

{EVENT.TIMESTAMP}

UNIX timestamp события, которое запустило действие.

{EVENT.VALUE}

Числовое значение события, которое запустило действие.
 Возможные значения: *1* — проблема, *0* — восстановление.

Макросы событий причины/симптома

{EVENT.CAUSE.ACK.STATUS}

Статус подтверждения события-причины (Да/Нет).
 Используется в контексте события-симптома.

{EVENT.CAUSE.AGE}

Возраст события-причины с точностью до секунды.
 Полезно в сообщениях с эскалацией.
 Используется в контексте события-симптома.

{EVENT.CAUSE.DATE}

Дата события-причины в формате `yyyy.mm.dd`.
 Используется в контексте события-симптома.

{EVENT.CAUSE.DURATION}

Длительность события-причины (разница во времени между событиями проблемы и восстановления) с точностью до секунды.
 Полезно в сообщениях о восстановлении проблемы.
 Используется в контексте события-симптома.

{EVENT.CAUSE.ID}

Числовой ID события-причины.
 Используется в контексте события-симптома.

{EVENT.CAUSE.NAME}

Имя события-причины.
 Используется в контексте события-симптома.

{EVENT.CAUSE.NSEVERITY}

Числовое значение важности события-причины.
 Возможные значения: *0* — Не классифицировано, *1* — Информация, *2* — Предупреждение, *3* — Средняя, *4* — Высокая, *5* — Катастрофа.
 Используется в контексте события-симптома.

{EVENT.CAUSE.OBJECT}

Числовое значение объекта события-причины.
 Возможные значения: *0* — триггер, *1* — обнаруженный узел сети, *2* — обнаруженный сервис, *3* — авторегистрация, *4* — элемент данных, *5* — правило обнаружения низкого уровня.
 Используется в контексте события-симптома.

{EVENT.CAUSE.OPDATA}

Оперативные данные базового триггера причинной проблемы.
 Используется в контексте события-симптома.

{EVENT.CAUSE.SEVERITY}

Название важности события-причины.
 Возможные значения: *Не классифицировано*, *Информация*, *Предупреждение*, *Средняя*, *Высокая*, *Катастрофическая*.
 Используется в контексте события-симптома.

{EVENT.CAUSE.SOURCE}

Числовое значение источника события-причины.
 Возможные значения: *0* — Триггер, *1* — Обнаружение, *2* — Авторегистрация, *3* — Внутреннее.
 Используется в контексте события-симптома.

{EVENT.CAUSE.STATUS}

Текстовое значение события-причины.
 Используется в контексте события-симптома.

{EVENT.CAUSE.TAGS}

Разделённый запятыми список тегов события-причины.
Разворачивается в пустую строку, если теги отсутствуют.
 Используется в контексте события-симптома.

{EVENT.CAUSE.TAGSJSON}

JSON-массив, содержащий объекты тегов причинного события.
Разворачивается в пустой массив, если теги отсутствуют.
 Используется в контексте события-симптома.

{EVENT.CAUSE.TAGS.<tag name>}

Значение тега события-причины, на которое ссылается имя тега.
 Имя тега, содержащее неалфавитно-цифровые символы (включая неанглийские многобайтовые UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные слэши внутри имени тега в кавычках должны быть экранированы обратным слэшем.
 Используется в контексте симптомного события.

{EVENT.CAUSE.TIME}

Время причинного события в формате `hh:mm:ss`.
 Используется в контексте события-симптома.

{EVENT.CAUSE.TIMESTAMP}

UNIX timestamp события-причины.
 Используется в контексте события-симптома.

{EVENT.CAUSE.UPDATE.HISTORY}

Журнал обновлений проблемы-причины (подтверждения и т. д.).
 Используется в контексте события-симптома.

{EVENT.CAUSE.VALUE}

Числовое значение события-причины.
 Возможные значения: *1* — Проблема, *0* — Восстановление.
 Используется в контексте события-симптома.

{EVENT.SYMPTOMS}

Список событий симптомов. Включает следующие сведения: имя узла сети, имя события, серьезность, возраст, теги службы и значения.
 Этот макрос используется в контексте события причины и возвращает информацию о событиях симптомов.

Макросы событий восстановления

{EVENT.RECOVERY.DATE}

Дата события восстановления в формате `yyyy.mm.dd`.

{EVENT.RECOVERY.ID}

Числовой ID события восстановления.

{EVENT.RECOVERY.NAME}

Имя события восстановления.

{EVENT.RECOVERY.STATUS}

Текстовое значение события восстановления.

{EVENT.RECOVERY.TAGS}

Список тегов события восстановления, разделенных запятыми. Если тегов нет, разворачивается в пустую строку.

{EVENT.RECOVERY.TAGSJSON}

JSON-массив, содержащий объекты тегов события восстановления. Расширяется до пустого массива, если теги отсутствуют.

{EVENT.RECOVERY.TIME}

Время события восстановления в формате `hh:mm:ss`.

{EVENT.RECOVERY.TIMESTAMP}

UNIX timestamp события восстановления.

{EVENT.RECOVERY.VALUE}

Числовое значение события восстановления.

Макросы обновления события

{EVENT.UPDATE.ACTION}

Читаемое человеком название действия(й), выполненного(ых) во время обновления проблемы.
 Принимает следующие значения: *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* и *closed* (в зависимости от того, сколько действий выполнено за одно обновление).

{EVENT.UPDATE.ACTIONJSON}

Массив JSON, содержащий сведения о действии(ях), выполненных во время обновления проблемы.
 Возможные значения свойств JSON:
- true (для свойств `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` и `symptom`);
- <строка сообщения> (для свойства `message`);
- timestamp (для свойств `suppress_until` и `timestamp`) или 0

(для свойства suppress_until, если подавление бессрочное);
- 0, 1, 2, 3, 4, 5 (для свойств серьезности old и new).

 Например: {"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"s

{EVENT.UPDATE.DATE}

Дата обновления события (подтверждение и т. д.) в формате yyyy.mm.dd.

{EVENT.UPDATE.HISTORY}

Журнал обновлений проблем (подтверждений и т. д.).

{EVENT.UPDATE.MESSAGE}

Сообщение обновления проблемы.

{EVENT.UPDATE.NSEVERITY}

Числовое значение нового уровня важности события, заданного во время операции обновления проблемы.

{EVENT.UPDATE.SEVERITY}

Имя новой важности события, заданной во время операции обновления проблемы.

{EVENT.UPDATE.STATUS}

Числовое значение статуса обновления проблемы.
 Возможные значения: *0* — вебхук был вызван из-за события проблемы/восстановления, *1* — операция обновления.

{EVENT.UPDATE.TIME}

Время обновления события (подтверждение и т. д.) в формате hh:mm:ss.

{EVENT.UPDATE.TIMESTAMP}

UNIX-метка времени обновления события (подтверждения и т. д.).

Функциональные макросы

{FUNCTION.VALUE}

Значение N-й функции, основанной на элементе данных, в выражении триггера на момент события.
 Учитываются только функции, у которых /host/key указан в качестве первого параметра.

 Этот макрос можно использовать с числовым индексом, например {FUNCTION.VALUE<1-9>}, чтобы указать на первую, вторую, третью и т. д. функцию в выражении триггера. См. indexed macros.

{FUNCTION.RECOVERY.VALUE}

Значение N-й функции на основе элемента данных в выражении восстановления на момент события.*
 Учитываются только функции с /host/key в качестве первого параметра.

 Этот макрос можно использовать с числовым индексом, например {FUNCTION.RECOVERY.VALUE<1-9>}, чтобы указать на первую, вторую, третью и т. д. функцию в выражении триггера. См. indexed macros.

Макросы узла сети

{HOST.CONN}

IP-адрес узла сети или DNS-имя, в зависимости от настроек узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет разрешаться в основной интерфейс агента. Если интерфейс агента не определен, будет использоваться основной SNMP-интерфейс. Если SNMP-интерфейс также не определен, будет использоваться основной JMX-интерфейс. Если JMX-интерфейс тоже не определен, будет использоваться основной IPMI-интерфейс. Если у узла сети нет ни одного интерфейса, макрос разрешается в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом в виде {HOST.CONN<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.DESCRIPTION}

Описание узла сети.

 Этот макрос может использоваться с числовым индексом в виде {HOST.DESCRIPTION<1-9>}, чтобы указывать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.DNS}

DNS-имя узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет разрешаться в основной интерфейс агента. Если интерфейс агента не определен, будет использоваться основной интерфейс SNMP. Если интерфейс SNMP также не определен, будет использоваться основной интерфейс JMX. Если интерфейс JMX тоже не определен, будет использоваться основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос разрешается в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом в виде {HOST.DNS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.HOST}

Техническое имя узла сети.
 Функции макросов не поддерживаются для этого макроса, если он используется как заполнитель в первом параметре функции истории, например, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Этот макрос может использоваться с числовым индексом в виде {HOST.HOST<1-9>} для указания на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.ID}

ID узла сети.

 Этот макрос можно использовать с числовым индексом в виде {HOST.ID<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.IP}

IP-адрес узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет подставляться для основного интерфейса агента. Если интерфейс агента не определен, будет использоваться основной интерфейс SNMP. Если интерфейс SNMP также не определен, будет использоваться основной интерфейс JMX. Если интерфейс JMX тоже не определен, будет использоваться основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос принимает значение 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом в виде {HOST.IP<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.METADATA}

Метаданные узла сети.

 Используется только для активной авторегистрации агента.

{HOST.NAME}

Отображаемое имя узла сети.

 Этот макрос можно использовать с числовым индексом в виде {HOST.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{HOST.PORT}

Порт узла сети (агент).
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет подставлять значение основного интерфейса агента. Если интерфейс агента не определен, будет использован основной интерфейс SNMP. Если интерфейс SNMP также не определен, будет использован основной интерфейс JMX. Если интерфейс JMX тоже не определен, будет использован основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос принимает значение 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом в виде {HOST.PORT<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

Макросы целевого узла сети

{HOST.TARGET.CONN}

IP-адрес или DNS-имя целевого узла сети.

{HOST.TARGET.DNS}

DNS-имя целевого узла сети.

{HOST.TARGET.HOST}

Техническое имя целевого узла сети.

{HOST.TARGET.IP}

IP-адрес целевого узла сети.

{HOST.TARGET.NAME}

Видимое имя целевого узла сети.

Макросы группы узлов сети

{HOSTGROUP.ID}

ID группы узлов сети.

Макросы инвентарных данных узла сети

{INVENTORY.ALIAS}

Поле *Alias* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.ALIAS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.ASSET.TAG}

Поле *Asset tag* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.ASSET.TAG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.CHASSIS}

Поле *Chassis* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.CHASSIS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.CONTACT}

Поле *Contact* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.CONTACT<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.CONTRACT.NUMBER}

Поле *Номер контракта* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.CONTRACT.NUMBER<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.DEPLOYMENT.STATUS}

Поле *Deployment status* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.DEPLOYMENT.STATUS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HARDWARE}

Поле *Hardware* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HARDWARE<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HARDWARE.FULL}

Поле *Аппаратное обеспечение (полные сведения)* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HARDWARE.FULL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HOST.NETMASK}

Поле *Маска подсети узла сети* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HOST.NETMASK<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HOST.NETWORKS}

Поле *Host networks* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HOST.NETWORKS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HOST.ROUTER}

Поле *маршрутизатор узла сети* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HOST.ROUTER<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HW.ARCH}

Поле *Аппаратная архитектура* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HW.ARCH<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HW.DATE.DECOMM}

Поле *Дата вывода оборудования из эксплуатации* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HW.DATE.DECOMM<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HW.DATE.EXPIRY}

Поле *Дата окончания срока аппаратного обслуживания* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HW.DATE.EXPIRY<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HW.DATE.INSTALL}

Поле *Дата установки оборудования* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HW.DATE.INSTALL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.HW.DATE.PURCHASE}

Поле *Дата покупки оборудования* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.HW.DATE.PURCHASE<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.INSTALLER.NAME}

Поле *Installer name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.INSTALLER.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.LOCATION}

Поле *Location* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.LOCATION<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.LOCATION.LAT}

Поле *Широта местоположения* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.LOCATION.LAT<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.LOCATION.LON}

Поле *Долгота местоположения* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.LOCATION.LON<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.MACADDRESS.A}

Поле *MAC address A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.MACADDRESS.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.MACADDRESS.B}

Поле *MAC address B* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.MACADDRESS.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.MODEL}

Поле *Model* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.MODEL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.NAME}

Поле *Name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.NOTES}

Поле *Notes* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.NOTES<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OOB.IP}

Поле *OOB IP address* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OOB.IP<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OOB.NETMASK}

Поле *маска подсети OOB* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OOB.NETMASK<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OOB.ROUTER}

Поле *OOB router* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OOB.ROUTER<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OS}

Поле *OS* в инвентаре узла сети.

 Этот макрос может использоваться с числовым индексом, например {INVENTORY.OS<1-9>}, чтобы указывать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OS.FULL}

Поле *OS (полные сведения)* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OS.FULL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.OS.SHORT}

Поле *OS (Short)* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OS.SHORT<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.CELL}

Поле *Основная ячейка POC* в инвентаре узла сети.

 Этот макрос может использоваться с числовым индексом, например {INVENTORY.POC.PRIMARY.CELL<1-9>}, чтобы указывать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.EMAIL}

Поле *Основной email POC* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{INVENTORY.POC.PRIMARY.NAME}

Поле *Primary POC name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.NOTES}

Поле *Примечания основного контактного лица* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.NOTES<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.PHONE.A}

Поле *Основной телефон POC A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.PHONE.B}

Поле *Primary POC phone B* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.PRIMARY.SCREEN}

Поле *Primary POC screen name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.CELL}

Поле *Secondary POC cell* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.CELL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.EMAIL}

Поле *Secondary POC email* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.NAME}

Поле *Secondary POC name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.NOTES}

Поле *Secondary POC notes* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.NOTES<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.PHONE.A}

Поле *Secondary POC phone A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.PHONE.B}

Поле *Secondary POC phone B* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.POC.SECONDARY.SCREEN}

Поле *Secondary POC screen name* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SERIALNO.A}

Поле *Серийный номер A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SERIALNO.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SERIALNO.B}

Поле *Серийный номер B* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SERIALNO.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.ADDRESS.A}

Поле *Адрес сайта A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.ADDRESS.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.ADDRESS.B}

Поле *Адрес сайта B* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.ADDRESS.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.ADDRESS.C}

Поле *Site address C* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.ADDRESS.C<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.CITY}

Поле *Город местоположения* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.CITY<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.COUNTRY}

Поле *Страна местоположения* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.COUNTRY<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.NOTES}

Поле *Примечания к сайту* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.NOTES<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{INVENTORY.SITE.RACK}

Поле *Site rack location* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.RACK<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.STATE}

Поле *State/Province* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.STATE<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SITE.ZIP}

Поле *Почтовый индекс/ZIP* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SITE.ZIP<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE}

Поле *Software* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.APP.A}

Поле *Software application A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.APP.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.APP.B}

Поле *Software application B* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.APP.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.APP.C}

Поле *Software application C* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.APP.C<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.APP.D}

Поле *Software application D* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.APP.D<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.APP.E}

Поле *Software application E* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.APP.E<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.SOFTWARE.FULL}

Поле *Software (Full details)* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.SOFTWARE.FULL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.TAG}

Поле *Tag* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.TAG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.TYPE}

Поле *Type* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.TYPE<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.TYPE.FULL}

Поле *Тип (полные сведения)* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.TYPE.FULL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{INVENTORY.URL.A}

Поле *URL A* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.URL.A<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.URL.B}

Поле *URL B* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.URL.B<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.URL.C}

Поле *URL C* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.URL.C<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{INVENTORY.VENDOR}

Поле *Vendor* в инвентаре узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.VENDOR<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

Макросы элементов данных

{ITEM.DESCRIPTION}

Описание N-го элемента данных в выражении триггера, которое вызвало уведомление.

 Этот макрос можно использовать с числовым индексом в виде {ITEM.DESCRIPTION<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{ITEM.DESCRIPTION.ORIG}

Исходное описание (с неразрешенными макросами) N-го элемента данных в выражении триггера, которое вызвало уведомление.

 Этот макрос можно использовать с числовым индексом в виде {ITEM.DESCRIPTION.ORIG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{ITEM.ID}

Числовой ID N-го элемента данных в выражении триггера, который вызвал уведомление.

 Этот макрос можно использовать с числовым индексом в виде {ITEM.ID<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{ITEM.KEY}

Ключ N-го элемента данных в выражении триггера, которое вызвало уведомление.
 Функции макросов не поддерживаются для этого макроса, если он используется как заполнитель в первом параметре функции истории, например, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Этот макрос может использоваться с числовым индексом в виде {ITEM.KEY<1-9>}, чтобы указывать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{ITEM.KEY.ORIG}

Исходный ключ (с неразрешенными макросами) N-го элемента данных в выражении триггера, который вызвал уведомление.

 Этот макрос можно использовать с числовым индексом в виде {ITEM.KEY.ORIG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. indexed macros.

{ITEM.LASTVALUE}

Последнее значение N-го элемента данных в выражении триггера, которое вызвало уведомление.
 В веб-интерфейсе оно будет разрешаться в *UNKNOWN*, если последнее значение из истории было собрано более чем *Max history display period* назад (задается в разделе меню *Administration > General*).
 При использовании в имени проблемы макрос не будет разрешаться в последнее значение элемента данных при просмотре событий проблемы; вместо этого он сохранит значение элемента данных на момент возникновения проблемы.
При использовании в уведомлениях в некоторых случаях макрос может не разрешаться в последнее значение элемента данных в момент срабатывания триггера. Например, если элемент данных быстро получает два значения, ”A” и ”B”, и триггер срабатывает для ”A”, в уведомлениях может отображаться ”B” как последнее значение из-за небольшой задержки обработки - последнее значение элемента данных изменилось между моментом срабатывания триггера и моментом создания уведомления. Чтобы избежать этого, можно использовать макрос {ITEM.VALUE}, который разрешается в значение на момент срабатывания триггера, обеспечивая использование правильного значения в уведомлении.
 Это

псевдоним для last(/{HOST.HOST}/{ITEM.KEY}).
 Разрешенное значение для текстовых/логовых элементов данных обрезается веб-интерфейсом до 20 символов в следующих местах:
- Оперативные данные;
- Описание триггера;
- URL триггера;
- Метки URL триггера;
- Описание виджета значения элемента данных.
Чтобы получить полное значение, можно использовать функции макросов, так как сервер не обрезает значения. Например: {{ITEM.LASTVALUE}.regsub("(.*)", \1)}.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LASTVALUE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.LASTVALUE.AGE}

Время, прошедшее между последним получением значения элемента данных и вычислением макроса.
 Отображается в удобочитаемом формате (например, 1m 45s).

 Этот макрос можно использовать с числовым индексом, например {ITEM.LASTVALUE.AGE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LASTVALUE.DATE}

Дата, когда было собрано последнее значение элемента данных.
 Отображается в формате YYYYMMDD.
 Часовой пояс сервера используется в названиях триггеров (в списке *Monitoring* > *Problems*), названиях событий, а также в названиях и значениях тегов. Во всех остальных случаях используется часовой пояс пользователя.

 Этот макрос может использоваться с числовым индексом, например, {ITEM.LASTVALUE.DATE<1-9>} для указания на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LASTVALUE.TIME}

Время, когда было собрано последнее значение элемента данных.
 Отображается в формате HHMMSS.
 Временная зона сервера используется в названиях триггеров (в списке *Monitoring* > *Problems*), названиях событий, а также в названиях и значениях тегов. Во всех остальных случаях используется временная зона пользователя.

 Этот макрос может использоваться с числовым индексом, например, {ITEM.LASTVALUE.TIME<1-9>} для указания на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LASTVALUE.TIMESTAMP}

Метка времени UNIX, когда было собрано последнее значение элемента данных.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LASTVALUE.TIMESTAMP<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LOG.AGE}

Возраст события журнала.
С точностью до секунды.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.AGE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LOG.DATE}

Дата, когда запись журнала была записана в журнал.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.DATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LOG.EVENTID}

ID события в журнале событий.
 Только для мониторинга журнала событий Windows.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.EVENTID<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LOG.NSEVERITY}

Числовая степень серьезности события в журнале событий.
 Только для мониторинга журнала событий Windows.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.NSEVERITY<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. indexed macros.

{ITEM.LOG.SEVERITY}

Словесная серьезность события в журнале событий.
 Только для мониторинга журнала событий Windows.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.SEVERITY<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. indexed macros.

{ITEM.LOG.SOURCE}

Источник события в журнале событий.
 Только для мониторинга журнала событий Windows.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.SOURCE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. indexed macros.

{ITEM.LOG.TIME}

Время, когда запись журнала была записана в журнал.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.TIME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.LOG.TIMESTAMP}

UNIX-метка времени, когда запись журнала была записана в журнал.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LOG.TIMESTAMP<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.NAME}

Имя элемента данных со всеми раскрытыми макросами.

 Этот макрос можно использовать с числовым индексом, например {ITEM.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.NAME.ORIG}

Исходное имя элемента данных (с неразрешенными макросами).

 Этот макрос можно использовать с числовым индексом, например {ITEM.NAME.ORIG<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.STATE}

Последнее состояние N-го элемента данных в выражении триггера, которое вызвало уведомление.
 Возможные значения: *Not supported*, *Normal*.

 Этот макрос можно использовать с числовым индексом, например {ITEM.STATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.STATE.ERROR}

Сообщение об ошибке с подробностями о том, почему элемент данных стал неподдерживаемым.
 Если элемент данных переходит в неподдерживаемое состояние, а затем сразу снова становится поддерживаемым, поле ошибки может быть пустым.

{ITEM.VALUE}

Разрешается в одно из следующих значений:

- Историческое значение N-го элемента данных в выражении триггера на момент события, если используется в контексте изменения состояния триггера, например при отображении событий или отправке уведомлений;
- Последнее значение N-го элемента данных в выражении триггера, если используется без контекста изменения состояния триггера, например при отображении списка триггеров во всплывающем окне выбора. В этом случае работает так же, как {ITEM.LASTVALUE}.

В первом случае будет разрешаться в *UNKNOWN*, если историческое значение уже было удалено или никогда не сохранялось.
Во втором случае, и только в веб-интерфейсе, будет разрешаться в *UNKNOWN*, если последнее историческое значение было собрано более чем *Max history display period* времени назад (задается в разделе меню *Administration > General*).
Разрешенное значение для текстовых/log-элементов данных обрезается веб-интерфейсом до 20 символов в следующих местах:
- Operational data;
- Trigger description;
- Trigger URLs;
- Trigger URL labels;
- Description of the item value widget.
Чтобы получить полное значение, можно использовать macro functions, так как сервер не обрезает значения. Например: `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Эта макрос может использоваться с числовым индексом, например {ITEM.VALUE<1-9>}, чтобы указывать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.VALUE.AGE}

Время, прошедшее между сбором значения элемента данных и вычислением макроса.
 Отображается в удобочитаемом формате (например, 1m 45s).

 Этот макрос можно использовать с числовым индексом, например {ITEM.VALUE.AGE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. indexed macros.

{ITEM.VALUE.DATE}

Дата, когда было собрано значение элемента данных.
 Отображается в формате `YYYYMMDD`.
 Временная зона сервера используется в названиях триггеров (в списке *Monitoring > Problems*), названиях событий, а также в названиях и значениях тегов. Во всех остальных случаях используется временная зона пользователя.

Этот макрос может использоваться с числовым индексом, например, {ITEM.VALUE.DATE<1-9>}, чтобы указывать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.VALUE.TIME}

Время, когда было собрано значение элемента данных.
 Отображается в формате HHMMSS.
 Часовой пояс сервера используется в именах триггеров (в списке *Monitoring* > *Problems*), именах событий, а также в именах и значениях тегов. Во всех остальных случаях используется часовой пояс пользователя.

Этот макрос может использоваться с числовым индексом, например, {ITEM.VALUE.TIME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.VALUE.TIMESTAMP}

Метка времени UNIX, когда было собрано значение элемента данных.

 Этот макрос можно использовать с числовым индексом, например {ITEM.VALUE.TIMESTAMP<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

{ITEM.VALUETYPE}

Тип значения N-го элемента данных в выражении триггера, которое вызвало уведомление.
 Возможные значения: *0* - Numeric float, *1* - Character, *2* - Log, *3* - Numeric unsigned, *4* - Text.

 Этот макрос можно использовать с числовым индексом, например {ITEM.VALUETYPE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. indexed macros.

Макросы уведомлений низкоуровневого обнаружения

{LLDRULE.DESCRIPTION}

Описание правила низкоуровневого обнаружения, которое вызвало уведомление.

{LLDRULE.DESCRIPTION.ORIG}

Исходное описание (с неразрешёнными макросами) правила низкоуровневого обнаружения, которое вызвало уведомление.

{LLDRULE.ID}

Числовой ID правила низкоуровневого обнаружения, вызвавшего уведомление.

{LLDRULE.KEY}

Ключ правила обнаружения низкого уровня, которое вызвало уведомление.

{LLDRULE.KEY.ORIG}

Исходный ключ (с неразрешёнными макросами) правила обнаружения низкого уровня, которое вызвало уведомление.

{LLDRULE.NAME}

Имя правила низкоуровневого обнаружения, вызвавшего уведомление.

{LLDRULE.NAME.ORIG}

Исходное имя (с неразрешёнными макросами) правила обнаружения низкого уровня, которое вызвало уведомление.

{LLDRULE.STATE}

Последнее состояние правила обнаружения низкого уровня.
 Возможные значения: *Not supported*, *Normal*.

 Этот макрос можно использовать с числовым индексом, например {LLDRULE.STATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. indexed macros.

{LLDRULE.STATE.ERROR}

Сообщение об ошибке с подробностями о том, почему правило обнаружения низкого уровня стало неподдерживаемым.
 Если правило обнаружения низкого уровня переходит в неподдерживаемое состояние, а затем сразу снова становится поддерживаемым, поле ошибки может быть пустым.

Макросы карты

{MAP.ID}

Идентификатор карты сети.

{MAP.NAME}

Имя карты сети.

Макросы прокси

{PROXY.DESCRIPTION}

Описание прокси.
 Разрешается в одно из следующих значений:

- Прокси N-го элемента данных в выражении триггера (в уведомлениях на основе триггеров). Здесь можно использовать макросы с индексом;

- Прокси, который выполнил обнаружение (в уведомлениях обнаружения). Здесь используйте {PROXY.DESCRIPTION} без индекса;
- Прокси, к которому зарегистрировался активный агент (в уведомлениях авто-регистрации). Здесь используйте {PROXY.DESCRIPTION} без индекса.

Этот макрос можно использовать с числовым индексом, например {PROXY.DESCRIPTION<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. макросы с индексом.

{PROXY.NAME}

Имя прокси.
 Разрешается в одном из следующих случаев:

- Прокси N-го элемента данных в выражении триггера (в уведомлениях на основе триггеров). Здесь можно использовать индексированные макросы;
- Прокси, который выполнил обнаружение (в уведомлениях обнаружения). Здесь используйте {PROXY.NAME} без индексации;
- Прокси, к которому зарегистрировался активный агент (в уведомлениях авто-регистрации). Здесь используйте {PROXY.NAME} без индексации.

Этот макрос можно использовать с числовым индексом, например {PROXY.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент в выражении триггера. См. индексированные макросы.

Макросы скрипта

{MANUALINPUT}

Значение ручного ввода, указанное пользователем во время выполнения скрипта.

Макросы сервиса

{SERVICE.DESCRIPTION}

Описание сервиса с раскрытыми макросами.

{SERVICE.ID}

Числовой идентификатор сервиса, который запустил действие.

{SERVICE.NAME}

Имя сервиса с раскрытыми макросами.

{SERVICE.ROOTCAUSE}

Список событий проблем триггеров, которые привели к отказу сервиса, отсортированный по важности и имени узла сети.
 Включает следующие сведения: имя узла сети, имя события, важность, возраст, теги сервиса и значения.

{SERVICE.TAGS}

Список тегов событий сервиса, разделенных запятыми.
 Теги событий сервиса можно определить в разделе тегов конфигурации сервиса. Если тегов нет, расширяется в пустую строку.

{SERVICE.TAGSJSON}

JSON-массив, содержащий объекты тегов событий сервиса.
 Теги событий сервиса можно определить в разделе тегов конфигурации сервиса. Если тегов нет, разворачивается в пустой массив.

{SERVICE.TAGS.<tag name>}

Значение тега события сервиса, на которое ссылается имя тега.
 Теги события сервиса можно определить в разделе тегов конфигурации сервиса.
 Имя тега, содержащее неалфавитно-цифровые символы (включая неанглийские многобайтовые UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные слэши внутри имени тега в кавычках должны быть экранированы обратным слэшем.

Макросы триггеров

{TRIGGER.DESCRIPTION}

Описание триггера.
 Все макросы, поддерживаемые в описании триггера, будут развернуты, если `{TRIGGER.DESCRIPTION}` используется в тексте уведомления.

{TRIGGER.EVENTS.ACK}

Количество подтверждённых событий для элемента карты в картах или для триггера, который сгенерировал текущее событие в уведомлениях.

{TRIGGER.EVENTS.PROBLEM.ACK}

Количество подтверждённых событий проблем для всех триггеров без учёта их состояния.

{TRIGGER.EVENTS.PROBLEM.UNACK}

Количество неподтверждённых событий проблем для всех триггеров без учёта их состояния.

{TRIGGER.EVENTS.UNACK}

Количество неподтверждённых событий для элемента карты в картах или для триггера, который сгенерировал текущее событие в уведомлениях.

{TRIGGER.EXPRESSION}

Выражение триггера.

{TRIGGER.EXPRESSION.EXPLAIN}

Частично вычисленное выражение триггера.
 Функции, основанные на элементах данных, вычисляются и заменяются результатами в момент создания события, тогда как все остальные функции отображаются в том виде, в котором они записаны в выражении. Может использоваться для отладки выражений триггера.

{TRIGGER.EXPRESSION.RECOVERY}

Выражение восстановления триггера, если *Генерация событий OK* в настройках триггера установлена в значение "Recovery expression"; в противном случае возвращается пустая строка.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Частично вычисленное выражение восстановления триггера.
 Функции, основанные на элементах данных, вычисляются и заменяются результатами в момент генерации события, тогда как все остальные функции отображаются так, как они записаны в выражении. Можно использовать для отладки выражений восстановления триггера.

{TRIGGER.HOSTGROUP.NAME}

Отсортированный (по SQL-запросу) список групп узлов сети, в которых определён триггер, разделённый запятыми и пробелами.

{TRIGGER.ID}

Числовой идентификатор триггера, который запустил это действие.

{TRIGGER.NAME}

Имя триггера с раскрытыми макросами.
 Обратите внимание, что `{EVENT.NAME}` можно использовать в действиях для отображения имени сработавшего события/проблемы с раскрытыми макросами.

{TRIGGER.NAME.ORIG}

Исходное имя триггера (с неразрешёнными макросами).

{TRIGGER.NSEVERITY}

Числовая важность триггера.
 Возможные значения: *0* — Не классифицировано, *1* — Информация, *2* — Предупреждение, *3* — Средняя, *4* — Высокая, *5* — Катастрофа.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

Количество подтвержденных событий проблем для триггеров в состоянии проблемы.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

Количество неподтверждённых событий проблем для триггеров в состоянии проблемы.

{TRIGGER.SEVERITY}

Имя важности триггера.
 Может быть задано в *Administration* > *General* > *Trigger displaying options*.

{TRIGGER.STATE}

Последнее состояние выражения триггера.
 Возможные значения: *Неизвестно*, *Нормально*.

{TRIGGER.STATE.ERROR}

Сообщение об ошибке с подробностями о том, почему триггер стал неподдерживаемым.
 Если триггер переходит в неподдерживаемое состояние, а затем сразу снова становится поддерживаемым, поле ошибки может быть пустым.

{TRIGGER.STATUS}

Значение триггера на момент выполнения шага операции.
 Возможные значения: *Проблема*, *OK*.

{TRIGGER.TEMPLATE.NAME}

Отсортированный (SQL-запросом) список шаблонов, в которых определён триггер, разделённый запятыми и пробелами, или *UNKNOWN*, если триггер определён на узле сети.

{TRIGGER.URL}

URL триггера.

{TRIGGER.URL.NAME}

Метка для URL триггера.

{TRIGGER.VALUE}

Текущее числовое значение триггера.
 Возможные значения: *0* — триггер находится в состоянии OK, *1* — триггер находится в состоянии Проблема.

{TRIGGERS.ACK}

Количество подтвержденных триггеров для элемента карты без учета состояния триггера.
 Триггер считается подтвержденным, если подтверждены все его события проблем.

{TRIGGERS.PROBLEM.ACK}

Количество подтвержденных проблемных триггеров для элемента карты.
 Триггер считается подтвержденным, если подтверждены все его события проблемы.

{TRIGGERS.PROBLEM.UNACK}

Количество неподтвержденных проблемных триггеров для элемента карты.
 Триггер считается неподтвержденным, если по крайней мере одно из его событий проблемы не подтверждено.

{TRIGGERS.UNACK}

Количество неподтверждённых триггеров для элемента карты, без учёта состояния триггера.
 Триггер считается неподтверждённым, если хотя бы одно из его событий проблемы не подтверждено.

Макросы имени пользователя

{USER.FULLNAME}

Имя, фамилия и имя пользователя пользователя, который добавил подтверждение события или запустил скрипт.

{USER.NAME}

Имя пользователя, который запустил скрипт.

{USER.SURNAME}

Фамилия пользователя, который запустил скрипт.

{USER.USERNAME}

Имя пользователя, который запустил скрипт.

Индексированные макросы

Синтаксис индексированного макроса **{MACRO<1-9>}** может ссылаться только на N-й элемент данных, функцию или узел сети в поле *Выражение* триггера:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** подставляют IP-адреса первого, второго и третьего узлов сети в выражении триггера (если они присутствуют).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** подставляют значения первого, второго и третьего элементов данных в выражении триггера на момент события (если они присутствуют).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** подставляют значения первой, второй и третьей функций, основанных на элементах данных, на момент события (если они присутствуют).

В контексте триггеров индексированные макросы всегда относятся к полю *Выражение* в конфигурации триггера, а не к *Выражению восстановления*. Например, в событии восстановления **{ITEM.VALUE2}** будет подставлять значение второго элемента данных из выражения проблемы на момент восстановления.

Макрос **{HOST.HOST<1-9>}** также поддерживается в макросе выражения `{?func(/host/key,param)}` в **именах графиков**. Например, `{?func(/{HOST.HOST2}/key,param)}` в имени графика будет подставлять узел сети второго элемента данных в графике.

Warning:
Индексированные макросы не будут подставляться ни в каком другом контексте, кроме случаев, перечисленных здесь. Для других контекстов используйте макросы без индекса (**{HOST.HOST}, {HOST.IP}** и т. д.).

Поддерживаемые пользовательские макросы по расположению

Обзор

В этом разделе представлен список мест, где поддерживаются определяемые пользователем макросы.

Note:
Для *Действий*, *Сетевого обнаружения*, *Прокси* и всех мест, перечисленных в разделе *Другие места* этой страницы, поддерживаются только пользовательские макросы, определённые на глобальном уровне. В упомянутых местах макросы, определённые на уровне узла сети и на уровне шаблона, раскрываться не будут.

Note:
Чтобы настроить значения макросов (например, сократить или извлечь определённые подстроки), можно использовать функции макросов.

Действия

При настройке действий пользовательские макросы могут быть использованы в следующих полях:

Место		Несколько макросов/вперемешку с
Оповещения и команды на события триггеров		да
Внутренние оповещения на события триггеров		да
Оповещения об обновлении проблем		да
Оповещения и команды на события услуг		да
Оповещения об обновлении услуг		да
Условие периода времени		нет
Операции		
	Длительность шага операции по умолчанию	нет
	Длительность шага	нет

Узлы сети / прототипы узлов сети

При настройке узла сети и прототипа узла сети пользовательские макросы могут использоваться в следующих полях:

Место		Несколько макросов/вперемешку с текстом[1]
IP адрес / DNS имя интерфейса		только DNS
Порт интерфейса		нет
SNMP v1, v2		
	SNMP community	да
SNMP v3		
	Имя контекста	да
	Имя безопасности	да
	Пароль аутентификации	да
	Ключевая фраза безопасности	да
IPMI		
	Имя пользователя	да
	Пароль	да
Теги[2]		
	Имена тегов	да
	Значения тегов	да

Элементы данных / прототипы элементов данных

В конфигурации элемента данных или прототипа элемента данных пользовательские макросы можно использовать в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Имя элемента данных	да
Параметры ключа элемента данных	да[4]
Интервал обновления	нет
Пользовательские интервалы	нет
Тайм-аут (доступен для поддерживаемых типов элементов данных)	нет
Хранить до (для истории и трендов)	нет
Описание	да
Вычисляемый/агрегированный элемент данных	
Формула (константы выражения и параметры функций; параметры ключа элемента данных; (*только для агрегированного элемента данных*) условия фильтрации (имя группы узлов сети и имя тега))	да
Монитор базы данных	
Имя пользователя	да
Пароль	да
SQL-запрос	да
HTTP-агент	
URL[3]	да
Поля запроса	да
Тело запроса	да
Заголовки (имена и значения)	да
Требуемые коды состояния	да
HTTP-прокси	да
Имя пользователя для HTTP-аутентификации	да
Пароль для HTTP-аутентификации	да
Файл сертификата SSl	да
Файл ключа SSl	да
Пароль ключа SSl	да
Разрешенные узлы сети	да
JMX-агент	
Конечная точка JMX	да
Элемент данных типа «Скрипт»	
Имена и значения параметров	да

Расположение	Несколько макросов/смешивание с текстом[1]
Элемент данных типа «Браузер»	
Имена и значения параметров	да
Элемент данных типа «Запрос телеметрии»	
Смещение времени	нет
Ограничение периода поиска	нет
Детализация	нет
SNMP-агент	
OID SNMP	да
SSH-агент	
Имя пользователя	да
Файл открытого ключа	да
Файл закрытого ключа	да
Пароль	да
Скрипт	да
TELNET-агент	
Имя пользователя	да
Пароль	да
Скрипт	да
Zabbix trapper	
Разрешенные узлы сети	да
Теги[2]	
Имена тегов	да
Значения тегов	да
Шаги предварительной обработки	
Параметры (включая пользовательские скрипты)	да
Пользовательские параметры обработки ошибок (поля *Установить значение* и *Установить ошибку в*)	да

Низкоуровневое обнаружение

При настройке правил низкоуровневого обнаружения пользовательские макросы могут использоваться в следующих полях:

Место	Несколько макросов/вперемешку с текстом[1]
Параметры ключа	да
Интервал обновления	нет
Пользовательские интервалы	нет

Место	Несколько макросов/вперемешку с текстом[1]
Время ожидания (доступно для поддерживаемых типов элементов данных)	нет
Удаление потерянных ресурсов	нет
Деактивация потерянных ресурсов	нет
Описание	да
SNMP агент	
SNMP OID	да
SSH агент	
Имя пользователя	да
Файл публичного ключа	да
Файл приватного ключа	да
Пароль	да
Выполняемый скрипт	да
TELNET агент	
Имя пользователя	да
Пароль	да
Выполняемый скрипт	да
Zabbix траппер	
Разрешённые хосты	да
Монитор баз данных	
Имя пользователя	да
Пароль	да
SQL запрос	да
JMX агент	
JMX endpoint	да
HTTP агент	
URL[3]	да
Поля запроса	да
Тело запроса	да
Заголовки (имена и значения)	да
Требуемые коды состояния	да
Имя пользователя для HTTP аутентификации	да
Пароль для HTTP аутентификации	да
Фильтры	
Регулярное выражение	да
Замещения	
Фильтры: регулярное выражение	да
Операции: интервал обновления (для прототипов элементов данных)	нет
Операции: период хранения истории (для прототипов элементов данных)	нет
Операции: период хранения динамики изменений (для прототипов элементов данных)	нет

Сетевое обнаружение

При настройке правил сетевого обнаружения пользовательские макросы могут использоваться в следующих полях:

Место		Несколько макросов/вперемешку с текстом[1]
Интервал обновления		нет
SNMP v1, v2		
	SNMP community	да
	SNMP OID	да
SNMP v3		
	Имя контекста	да
	Имя безопасности	да
	Пароль аутентификации	да
	Ключевая фраза безопасности	да
	SNMP OID	да

Прокси

В конфигурации прокси пользовательские макросы можно использовать в следующих полях:

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the proxy belongs to a group)	no
Interface address and port (for passive proxy)	no
Timeouts for item types	no

Группы прокси

В конфигурации группы прокси пользовательские макросы можно использовать в следующих полях:

Местоположение	Несколько макросов/смешивание с текстом[1]
Период переключения на резервный прокси	нет
Минимальное количество прокси	нет

Шаблоны

При настройке шаблонов пользовательские макросы могут использоваться в следующих полях:

Место		Несколько макросов/вперемешку с текстом[1]
Теги[2]		
	Имена тегов	да
	Значения тегов	да

Триггеры

При настройке триггеров пользовательские макросы могут использоваться в следующих полях:

Место	Несколько макросов/вперемешку с текстом[1]
Имя	да
Оперативные данные	да

Место	Несколько макросов/вперемешку с текстом[1]
Выражение (только в константах и параметрах функций; секретные макросы не поддерживаются)	да
Теги для совпадения	да
Имя записи в меню	да
URL записи в меню[3]	да
Описание	да
Теги[2]	
Имена тегов	да
Значения тегов	да

Веб-сценарии

При настройке веб-сценариев пользовательские макросы могут использоваться в следующих полях:

Место		Несколько макросов/вперемешку с текстом[1]
Имя		да
Интервал обновления		нет
Агент		да
HTTP прокси		да
Переменные (только значения)		да
Заголовки (имена и значения)		да
Шаги		
	Имя	да
	URL[3]	да
	Переменные (только значения)	да
	Заголовки (имена и значения)	да
	Время ожидания	нет
	Требуемая строка	да
	Требуемые коды состояния	нет
Аутентификация		
	Пользователь	да
	Пароль	да
	Файл SSL сертификата	да
	Файл SSL ключа	да
	Пароль к SSL ключу	да
Теги[2]		
	Имена тегов	да
	Значения тегов	да

Другие расположения

Помимо расположений, перечисленных здесь, пользовательские макросы можно использовать в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Глобальные скрипты (URL, script, SSH, Tel-net, IPMI), включая текст подтверждения	yes
Вебхуки	
Скрипт JavaScript	no
Имя параметра скрипта JavaScript	no
Значение параметра скрипта JavaScript	yes
Панели	
Столбец типа данных *Text* в виджете панели *Top hosts*	yes
Параметр *Description* в виджете панели *Item value* и *Gauge*	yes
Параметр *Text* для основной/вторичной метки в виджете панели *Honeycomb*	yes
Параметр *URL*[3] в виджете панели *URL*	yes
Мониторинг > *Проблемы*, *Узлы сети*, *Последние данные*	
Столбец *Custom text*	yes
Сбор данных > *Шаблоны*, *Узлы сети*	
Столбец *Custom text*	yes
Оповещения > *Типы медиа* > *Шаблоны сообщений*	
Тема	yes
Сообщение	yes
Оповещения > *Типы медиа* > *Скрипт*	
Параметры скрипта	yes
Оповещения > *Типы медиа* > *Тип медиа*	

Расположение	Несколько макросов/смешивание с текстом[1]
Поля *Username* и *Password* для типа медиа *Email* (когда для *Authentication* установлено значение "Username and password"; рекомендуется использовать секретные макросы) *Пользователи* > *Пользователи* > *Медиа*	yes
Когда активно *Администрирование* > *Общие* > *GUI*	yes
Рабочее время *Администрирование* > *Общие* > *Тайм-ауты*	no
Тайм-ауты для типов элементов данных *Администрирование* > *Общие* > *Коннекторы*	no
URL	yes
Имя пользователя	yes
Пароль	yes
Bearer token	yes
Тайм-аут	no
HTTP proxy	yes
Файл SSL-сертификата	yes
Файл SSL-ключа	yes
Пароль SSL-ключа	yes

Полный список всех макросов, поддерживаемых в Zabbix, см. в разделе поддерживаемые макросы.

Сноски

[1] Если для поля не поддерживаются несколько макросов или макросы, смешанные с текстом, для указанного места, одно значение макроса должно заполнять все поле.

[2] Макросы, используемые в именах и значениях тегов, разрешаются только в процессе генерации события.

[3] URL, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешен как "******".

[4] Макросы не поддерживаются в ключе элемента данных Zabbix trapper.

Совместимость версий

Поддерживаемые агенты

Для совместимости с Zabbix 8.0 версия агента Zabbix должна быть не ниже 1.6 и не выше 8.0.

Возможно, потребуется проверить конфигурацию более старых агентов, поскольку некоторые параметры изменились. Например, это касается параметров, связанных с журналированием, в версиях до 3.0.

Чтобы в полной мере использовать новейшие функции, метрики, повышенную производительность и сниженное потребление памяти, используйте последнюю поддерживаемую версию агента.

Примечания для Windows XP

- В 32-разрядной версии Windows XP не используйте агенты Zabbix новее 6.0.x;
- В Windows XP/Server 2003 не используйте шаблоны агентов новее Zabbix 4.0.x. В более новых шаблонах используются счетчики производительности на английском языке, которые поддерживаются только начиная с Windows Vista/Server 2008.

Поддерживаемые агенты 2

Более старые версии Zabbix agent 2, начиная с версии 4.4, совместимы с Zabbix 8.0; Zabbix agent 2 не должен быть новее 8.0.

Обратите внимание, что при использовании Zabbix agent 2 версий 4.4 и 5.0 для обновления неподдерживаемых элементов данных используется интервал по умолчанию 10 минут.

Чтобы в полной мере воспользоваться последними функциями, метриками, улучшенной производительностью и сниженным потреблением памяти, используйте последнюю поддерживаемую версию agent 2.

Поддерживаемые прокси Zabbix

Для полной совместимости с Zabbix 8.0 прокси должны соответствовать основной версии сервера. Только прокси Zabbix 8.0.x полностью совместимы с сервером Zabbix 8.0.x.

Устаревшие прокси поддерживаются частично: они по-прежнему могут собирать данные и выполнять скрипты, но не могут получать обновления конфигурации, такие как новые элементы данных.

В отношении сервера Zabbix прокси могут быть:

- *Текущими* (прокси и сервер имеют одну и ту же основную версию);
- *Устаревшими* (версия прокси старее версии сервера, но частично поддерживается);
- *Неподдерживаемыми* (версия прокси старее версии предыдущего LTS-релиза сервера *или* версия прокси новее основной версии сервера).

Примеры:

Версия сервера	*Текущая* версия прокси	*Устаревшая* версия прокси	*Неподдерживаемая* версия прокси
6.4	6.4	6.0, 6.2	Старее 6.0; новее 6.4
7.0	7.0	6.0, 6.2, 6.4	Старее 6.0; новее 7.0
7.2	7.2	7.0	Старее 7.0; новее 7.2
7.4	7.4	7.0	Старее 7.0; новее 7.4
8.0	8.0	7.0, 7.2, 7.4	Старее 7.0; новее 8.0

Поддерживаемая прокси функциональность:

Версия прокси	Обновление данных	Обновление конфигурации	Задачи
Текущая	Да	Да	Да
Устаревшая	Да	Нет	Удаленные команды (например, сценарии shell); Немедленные проверки значений элементов данных (то есть *Выполнить сейчас*); Примечание: Предварительная обработка тестов с реальным значением не поддерживается.
Неподдерживаемая	Нет	Нет	Нет

Оповещения об использовании несовместимых версий демона Zabbix будут журналироваться.

Поддерживаемые XML-файлы

Для импорта в Zabbix 8.0 поддерживаются XML-файлы не старше версии 1.8.

Attention:
В формате XML-экспорта зависимости триггеров хранятся только по имени. Если существует несколько триггеров с одинаковым именем (например, с разными уровнями серьезности и выражениями), между которыми определена зависимость, импортировать их невозможно. Такие зависимости необходимо вручную удалить из XML-файла и добавить заново после импорта.

Синтаксис временного периода

Обзор

Чтобы указать период времени, должен использоваться следующий формат:

`d-d,hh:mm-hh:mm`

где эти символы означают следующее:

Символ	Описание
d	День недели: 1 — понедельник, 2 — вторник ,... , 7 — воскресенье
hh	Часы: 00-24
mm	Минуты: 00-59

Вы можете указать более одного периода времени, используя разделитель в виде точки с запятой (;):

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Если оставить период времени пустым, это будет эквивалентно 1-7,00:00-24:00, что является значением по умолчанию.

Attention:
Верхняя граница периодов времени не включается. Таким образом, если вы укажете период 09:00-18:00, то последней секундой в периоде будет 17:59:59.

Примеры

Рабочее время. С понедельника по пятницу с 9:00 до 18:00:

`1-5,09:00-18:00`

Рабочие часы плюс выходные. С понедельника по пятницу с 9:00 до 18:00, а также суббота и воскресенье с 10:00 до 16:00:

`1-5,09:00-18:00;6-7,10:00-16:00`

Обозначения единиц измерения

Обзор

Работа с большими значениями, такими как «86400», «104857600» или «1000000», может быть сложной и может приводить к ошибкам. Для упрощения настройки и улучшения читабельности Zabbix поддерживает символы единиц измерения (суффиксы), которые выполняют функцию множителей значений.

Использование суффиксов может упростить, например, настройку выражений триггеров, делая их более простыми для понимания и поддержки.

Выражения триггера без суффиксов:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Выражения триггера с суффиксами:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

Суффиксы также могут упростить настройку других объектов — таких как ключи элементов данных, виджеты и т. д, и помочь отобразить значения элементов данных в удобном для восприятия формате.

Note:
Чтобы узнать, поддерживает ли поле конфигурации объекта суффиксы, всегда смотрите соответствующую страницу для настраиваемого оъекта.

Суффиксы времени

Веб-интерфейс Zabbix поддерживает следующие суффиксы времени в конфигурации сущностей:
 s - секунды *(при использовании работает так же, как и необработанное значение)*;
 m - минуты;
 h - часы;
 d - дни;
 w - недели;
 M - месяцы *(только для функций трендов)*;
 y - годы *(только для функций трендов)*.

Суффиксы времени поддерживаются только для целых чисел. Например, 1h поддерживается, но 1,5h или 1.5h не поддерживается; вместо этого используйте 90m.

Суффиксы размера памяти

Zabbix поддерживает следующие суффиксы размера памяти:
 K — килобайты
 M — мегабайты
 G — гигабайты
 T — терабайты

Суффиксы значений элементов данных

Суффиксы также могут использоваться для отображения числовых значений элементов данных в удобном для восприятия формате.

Чтобы включить это, используйте один из следующих суффиксов в поле *Единицы измерения* при настройке элемента данных:
 B — байты;
 Bps — байты в секунду;
 s — секунды, отображаются с использованием до трёх самых больших ненулевых единиц времени;
 uptime — прошедшее время в формате «чч:мм:сс» или «N дней, чч:мм:сс»;
 unixtime — отметка времени Unix, отформатированная как «гггг.мм.дд чч:мм:сс».

К тому, каким образом эти суффиксы интерпретируются и отображаются, применяются следующие дополнительные правила :

- Для «B» (байты) и «Bps» (байты в секунду) Zabbix использует преобразование с основанием 2 (1K = 1024B), следуя стандарту JEDEC [en].
- Для остальных единиц (таких как Гц, Вт и т. д.) Zabbix использует преобразование по основанию 10 (1K = 1000).
- Для «s» (секунды):
  - Формат включает «ггг ммм ддд ччч ммм ссс мс»; отображаются только до трёх самых больших ненулевых единиц времени (например, «1M 10д 4ч»).
  - Если единица равна нулю и находится между двумя ненулевыми единицами, она опускается (например, «10д 56м» вместо «10д 0ч 56м»).

При использовании *Единиц измерения* к значениям элементов данных автоматически применяются следующие суффиксы множителей:
 K, **M**, **G**, **T** — кило, мега, гига, тера;
 P, **E**, **Z**, **Y** — пета, экса, зетта, иотта *(эти применяются только в веб-интерфейсе)*.

Чтобы предотвратить преобразование единиц измерения, используйте префикс ! (например, «!B» или «!s»).

В следующих примерах показано, как полученные значения элементов данных преобразуются на основе указанных единиц измерения:

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016
3470400 s → 1  10  4
2606400 s → 1  4
2592000 s → 1
2592001 s → 1
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1    , 00:00:00
881764 uptime → 10    , 04:56:04
32417764 uptime → 375    , 04:56:04
```

```
32417764 !uptime → 32417764 uptime

881764 unixtime → 1970-01-11 04:56:04 AM

17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
До версии Zabbix 4.0 существовал жёстко закодированный чёрный список единиц измерения, который состоял из `ms`, `rpm`, `RPM`, `%`. Этот чёрный список считается устаревшим, поэтому правильным способом предотвратить преобразование подобных единиц измерения является `!ms`, `!rpm`, `!RPM`, `!%`.

Выполнение команд

Zabbix использует общую функциональность для внешних проверок, пользовательских параметров, элементов данных system.run, пользовательских скриптов оповещений, удаленных команд и глобальных скриптов.

Этапы выполнения

Note:
По умолчанию, все скрипты в Zabbix выполняются с имользованием оболочки *sh*, и оболочку по умолчанию изменить нельзя. Чтобы использовать другую оболочку, можно использовать следующий приём: создать файл скрипта и вызвать этот скрипт при выполнении команды.

Команда/скрипт выполняется одинаково как на Unix, так и на Windows платформах:

1. Zabbix (родительский процесс) создаёт конвейер для связи
2. Zabbix настраивает данный конвейер для вывода данных от создаваемого дочернего процесса
3. Zabbix создаёт дочерний процесс (выполняет команду/скрипт)
4. Создается новая группа процесса (в Unix) или задача (в Windows) для дочернего процесса
5. Zabbix считывает из конвейера данные до тех пор, пока не истечёт установленное время ожидания или до момента, когда прекратится запись с другой стороны (закрыты ВСЕ обработчики/файловые дескрипторы). Обратите внимание, что дочерний процесс может создать еще некоторое количество процессов и выйти до того, как эти процессы завершатся или будет закрыт обработчик/файловый дескриптор.
6. Если установленное время ожидания не было достигнуто, Zabbix ждёт завершения начального дочернего процесса или ждёт достижения установленного времени ожидания.
7. Если начальный дочерний процесс завершил свою работу и время ожидания не вышло, тогда Zabbix проверяет код завершения начального дочернего процесса и сравнивает его с 0 (ненулевое значение считается ошибкой выполнения, только для пользовательских скриптов оповещений, удалённых команд и пользовательских скриптов, выполняемых на Zabbix сервере и Zabbix прокси)
8. На этот момент подразумевается, что всё выполнено, и всё дерево процессов (т.е. группа процессов или задача) завершается.

Attention:
Zabbix предполагает, что команда/скрипт завершили обработку, в тот момент, когда завершился изначальный дочерний процесс И никакие другие процессы всё ещё не держат открытым обработчик/файловый дескриптор вывода. Когда обработка завершена, ВСЕ созданные процессы завершаются.

Все двойные кавычки и обратная косая черта в команде экранируются обратной косой чертой, и вся команда заключается в двойные кавычки.

Проверка кода выхода

Код выхода проверяется по следующим условиям:

- Только для пользовательских скриптов оповещений, удаленных команд и пользовательских скриптов, выполняемых на сервере Zabbix и прокси Zabbix.

- Любой код выхода, отличный от 0, считается ошибкой выполнения.
- Содержимое стандартного потока ошибок и стандартного потока вывода для неудачных выполнений собирается и доступно во веб-интерфейсе (где отображается результат выполнения).
- Дополнительная запись в журнале может быть создана для удаленных команд, выполняемых на агенте/прокси Zabbix, путем включения параметра LogRemoteCommands на agent/proxy.

Возможные сообщения веб-интерфейса и записи журнала для неудачных команд/скриптов:

- Содержимое стандартного потока ошибок и стандартного потока вывода для неудачных выполнений (если есть).
- "Process exited with code: N." (для пустого вывода и кода выхода, не равного 0).
- "Process killed by signal: N." (если процесс завершен сигналом, только в Linux).
- "Process terminated unexpectedly." (если процесс завершен по неизвестным причинам).

Смотрите также

- Внешние проверки
- Пользовательские параметры
- Элементы данных system.run
- Пользовательские скрипты оповещений
- Удалённые команды
- Глобальные скрипты

Примеры экранирования

Обзор

На этой странице представлены примеры правильного экранирования при использовании регулярных выражений в различных контекстах.

> **Note:**
> При использовании конструктора триггерных выражений правильное экранирование в регулярных выражениях добавляется автоматически.

Примеры

Пользовательский макрос с контекстом

Регулярное выражение: `\.+\"[a-z]+<br>` Пользовательский макрос с контекстом: `{$MACRO:regex:"\.+\\"[a-z]+"}`

Замечания:

- обратные слэши не экранируются;
- кавычки экранируются.

Функция макросов в параметре ключа элемента данных

Регулярное выражение: `.+:(\d+)$<br>` Ключ элемента данных: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$"`

Замечания: - регулярное выражение внутри функции макросов `regsub` заключается в двойные кавычки (из-за того, что содержит закрывающие скобки); - Символы кавычек, обрамляющие регулярное выражение, экранируются (из-за того, что весь третий параметр элемента данных заключается в кавычки); - третий параметр ключа элемента данных заключается в кавычки, поскольку он содержит запятую.

Функция макросов низкоуровневого обнаружения (LLD)

Регулярное выражение: `\.+\"([a-z]+)<br>` Макрос LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Замечания:

- обратные слэши не экранируются;
- кавычки экранируются.

Функция макросов LLD внутри контекста пользовательского макроса

Регулярное выражение: `\.+\"([a-z]+)<br>` Макрос LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Пользовательский макрос с контекстом: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Замечания:

- экранирование обратных слэшей для LLD не меняется;
- при вставке макроса LLD в контекст пользовательского макроса нам нужно поместить это в строку:

1. Выражение макроса заключается в кавычки;
2. Кавычки экранируются; всего добавляется 3 новых обратных слэша.

Строковый параметр функции (любой)

В качестве примера используется `concat`.

Содержимое строки: `\.+\"[a-z]+<br>` Выражение: `concat("abc", "\\.\\\"[a-z]+")`

Замечание:

- Строковый параметр требует экранирования как для обратных слэшей, так и для кавычек.

Функция макросов LLD внутри строкового параметра функции

Регулярное выражение: `\.+\"([a-z]+)<br>` Макрос LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Выражение: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Замечания:

- Строковые параметры требуют экранирования как для обратных слэшей, так и для кавычек;
- Добавляется ещё один уровень экранирования, так как макрос будет раскрыт только после раскрытия кавычек в строке;

Пользовательский макрос с контекстом внутри строкового параметра функции

Регулярное выражение: `\.+\"[a-z]+<br>` Пользовательский макрос с контекстом: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Выражение: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Замечания:

- Как и в предыдущем примере, необходим дополнительный уровень экранирования;
- Обратные слэши и кавычки экранируются только для экранирования верхнего уровня (поскольку это строковый параметр).

Функция макросов LLD внутри контекста пользовательского макроса внутри функции

Регулярное выражение: `\.+\"([a-z])+<br>` Макрос LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Пользовательский макрос с контекстом: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Выражение: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Замечания по трём уровням экранирования:

1. Для функции макросов LLD, без экранирования обратных слэшей;
2. Для пользовательского макроса с контекстом, без экранирования обратных слэшей;
3. Для строкового параметра функции, с экранированием обратных слэшей.

Пользовательский макрос с контекстом просто внутри строки

Регулярное выражение: `\.+\"[a-z]+<br>` Пользовательский макрос с контекстом: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Внутри строки некоторого выражения, например: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Замечания:

- Строки также требуют экранирования обратных слэшей;
- Строки также требуют экранирования кавычек;
- Снова случай с двумя уровнями экранирования:

1. Экранирование для контекста пользовательского макроса — без экранирования обратных слэшей;
2. Экранирование, поскольку это строка — с экранированием обратных слэшей.

Что такое Zabbix

Zabbix был создан Алексеем Владышевым и в настоящее время активно разрабатывается и поддерживается компанией Zabbix SIA.

Zabbix — это распределённое решение для мониторинга корпоративного уровня с открытым исходным кодом.

Zabbix — это программное обеспечение, которое отслеживает многочисленные параметры сети, а также состояние и целостность серверов, виртуальных машин, приложений, служб, баз данных, веб-сайтов, облачных сред и многого другого. Zabbix использует гибкий механизм уведомлений, позволяющий пользователям настраивать оповещения по электронной почте практически для любого события. Это обеспечивает быструю реакцию на проблемы с серверами.

Zabbix предоставляет широкие возможности для создания отчётов и визуализации данных на основе сохранённых данных. Это делает Zabbix идеальным решением для планирования ресурсов.

Zabbix поддерживает как опрос, так и получение ловушек. Доступ ко всем отчётам и статистике Zabbix, а также к параметрам конфигурации осуществляется через веб-интерфейс. Веб-интерфейс позволяет оценивать состояние сети и работоспособность серверов из любого места. При правильной настройке Zabbix может играть важную роль в мониторинге ИТ-инфраструктуры. Это в равной степени относится как к небольшим организациям с несколькими серверами, так и к крупным компаниям с множеством серверов.

Zabbix доступен бесплатно. Zabbix написан и распространяется по лицензии AGPL-3.0. Это означает, что его исходный код свободно распространяется и доступен широкой публике.

Коммерческая поддержка доступна и предоставляется компанией Zabbix и её партнёрами по всему миру.

Узнайте больше о возможностях Zabbix.

Пользователи Zabbix Многие организации разного размера по всему миру полагаются на Zabbix как на основную платформу мониторинга.

Архитектура Zabbix состоит из нескольких основных программных компонентов. Их обязанности описаны ниже.

Сервер

Сервер Zabbix — это центральный компонент, которому агенты передают информацию о доступности и целостности, а также статистические данные. Сервер является центральным хранилищем, в котором сохраняются все конфигурационные, статистические и операционные данные.

Хранение данных в базе данных

Вся информация о конфигурации, а также данные, собранные Zabbix, хранятся в базе данных.

Веб-интерфейс

Для удобного доступа к Zabbix из любого места и с любой платформы предусмотрен веб-интерфейс. Интерфейс является частью сервера Zabbix и обычно (но не обязательно) работает на той же физической машине, что и сервер.

Прокси

Прокси Zabbix может собирать данные о производительности и доступности от имени сервера Zabbix. Прокси является необязательной частью развертывания Zabbix; однако он может быть очень полезен для распределения нагрузки одного сервера Zabbix.

Агент

Агенты Zabbix развертываются на объектах мониторинга для активного контроля локальных ресурсов и приложений и передачи собранных данных на сервер Zabbix. Начиная с Zabbix 4.4 доступны два типа агентов: Zabbix agent (легковесный, поддерживается на многих платформах, написан на C) и Zabbix agent 2 (более гибкий, легко расширяемый с помощью плагинов, написан на Go).

Поток данных Кроме того, важно сделать шаг назад и взглянуть на общий поток данных в Zabbix. Чтобы создать элемент данных, который собирает данные, сначала необходимо создать узел сети. Если рассматривать другую сторону спектра Zabbix, то для создания триггера сначала нужен элемент данных. Чтобы создать действие, у вас должен быть триггер. Таким образом, если вы хотите получать оповещение о том, что загрузка CPU на *Server X* слишком высока, сначала необходимо создать запись узла сети для *Server X*, затем элемент данных для мониторинга его CPU, затем триггер, который активируется при слишком высокой загрузке CPU, а затем действие, которое отправляет вам письмо. Хотя это может показаться большим количеством шагов, при использовании шаблонов на самом деле это не так. Однако благодаря такому дизайну можно создать очень гибкую конфигурацию.

Возможности Zabbix — это высокоинтегрированное решение для мониторинга сети, предлагающее множество возможностей в одном пакете.

Сбор данных

- проверки доступности и производительности
- поддержка SNMP (как trapping, так и polling), IPMI, JMX, мониторинга VMware
- пользовательские проверки
- сбор нужных данных через настраиваемые интервалы
- выполняется сервером/прокси и агентами

Гибкие определения порогов

- вы можете задавать очень гибкие пороги проблем, называемые триггерами, ссылаясь на значения из базы данных backend

Высоконастраиваемое оповещение

- отправку уведомлений можно настраивать для расписания эскалации, получателя и типа носителя
- уведомления можно сделать понятными и полезными с помощью макропеременных
- автоматические действия включают удаленные команды

Построение графиков в реальном времени

- по контролируемым элементам данных сразу строятся графики с использованием встроенной функции построения графиков

Возможности веб-мониторинга

- Zabbix может проходить по пути имитируемых щелчков мышью на веб-сайте и проверять функциональность и время отклика

Широкие возможности визуализации

- возможность создавать пользовательские графики, которые могут объединять несколько элементов данных в одном представлении
- карты сети
- слайд-шоу в виде обзора в стиле панели
- отчеты
- высокоуровневое (бизнес) представление контролируемых ресурсов

Хранение исторических данных

- данные хранятся в базе данных
- настраиваемая история
- встроенная процедура очистки

Простая настройка

- добавляйте контролируемые устройства как узлы сети
- узлы сети начинают мониторинг после добавления в базу данных
- применяйте шаблоны к контролируемым устройствам

Использование шаблонов

- группировка проверок в шаблонах
- шаблоны могут наследовать другие шаблоны

Обнаружение сети

- автоматическое обнаружение сетевых устройств
- авто-регистрация агента
- обнаружение файловых систем, сетевых интерфейсов и SNMP OID

Быстрый веб-интерфейс

- веб-интерфейс на PHP
- доступен откуда угодно
- можно переходить по нему кликами
- журнал аудита

Zabbix API

- Zabbix API предоставляет программируемый интерфейс к Zabbix для массовых операций, интеграции со сторонним ПО и других целей.

Система разрешений

- безопасная аутентификация пользователей
- для некоторых пользователей можно ограничить доступ к определенным представлениям

Полнофункциональный и легко расширяемый агент

- развертывается на контролируемых объектах
- может быть развернут как в Linux, так и в Windows

Двоичные демоны

- написаны на C для высокой производительности и небольшого потребления памяти

- легко портируются

Готовность к сложным средам

- удаленный мониторинг упрощается с использованием прокси Zabbix

Глоссарий В этом разделе вы можете узнать значение некоторых терминов, часто используемых в Zabbix.

узел сети

- любое физическое или виртуальное устройство, приложение, служба или любая другая логически связанная совокупность контролируемых параметров.

группа узлов сети

- логическое объединение узлов сети. Группы узлов сети используются при назначении прав доступа к узлам сети для разных групп пользователей.

элемент данных

- определенная часть данных, которую необходимо получить от узла сети, то есть метрика данных.

предобработка значений

- преобразование полученного значения метрики перед сохранением в базу данных.

триггер

- логическое выражение, определяющее порог проблемы и используемое для «оценки» данных, полученных в элементах данных.

Когда полученные данные превышают порог, триггеры переходят из состояния *OK* в состояние *Проблема*. Когда полученные данные ниже порога, триггеры остаются в состоянии *OK* или возвращаются в него.

шаблон

- набор сущностей (элементов данных, триггеров, графиков, правил обнаружения низкоуровневых сущностей, веб-сценариев), готовых к применению к одному или нескольким узлам сети.

Шаблоны предназначены для ускорения развертывания задач мониторинга на узле сети, а также для упрощения массового изменения задач мониторинга. Шаблоны напрямую связываются с отдельными узлами сети.

группа шаблонов

- логическое объединение шаблонов. Группы шаблонов используются при назначении прав доступа к шаблонам для разных групп пользователей.

событие

- единичное возникновение чего-либо, заслуживающего внимания, например изменение состояния триггера или выполнение обнаружения/автоматической регистрации агента.

тег события

- предварительно определенный маркер события. Он может использоваться для корреляции событий, детальной настройки разрешений и т. д.

корреляция событий

- метод гибкого и точного сопоставления проблем с их устранением.

Например, можно определить, что проблема, о которой сообщает один триггер, может быть устранена другим триггером, который даже может использовать другой метод сбора данных.

проблема

- триггер, находящийся в состоянии «Проблема».

обновление проблемы

- предоставляемые Zabbix возможности управления проблемами, такие как добавление комментария, подтверждение, изменение важности или закрытие вручную.

действие

- предварительно определенный способ реагирования на событие.

Действие состоит из операций (например, отправки уведомления) и условий (*когда* выполняется операция).

эскалация

- пользовательский сценарий выполнения операций в рамках действия; последовательность отправки уведомлений/выполнения удаленных команд.

макрос

- заполнитель переменной, который преобразуется в определенное значение в зависимости от контекста.

медиа

- средство доставки уведомлений; канал доставки.

уведомление

- сообщение о некотором событии, отправленное пользователю через выбранный канал медиа.

удаленная команда

- предварительно определенная команда, автоматически выполняемая на контролируемом узле сети при выполнении определенного условия.

веб-сценарий

- один или несколько HTTP-запросов для проверки доступности веб-сайта.

веб-интерфейс

- веб-интерфейс, предоставляемый Zabbix.

панель

- настраиваемый раздел веб-интерфейса, отображающий сводную информацию и визуализации важных данных в визуальных элементах, называемых виджетами.

виджет

- визуальный элемент, отображающий информацию определенного типа и источника (сводку, карту, график, часы и т. д.), используемый на панели.

Zabbix API

- Zabbix API позволяет использовать протокол JSON RPC для создания, обновления и получения объектов Zabbix (например, узлов сети, элементов данных, графиков и других объектов), а также для выполнения любых других пользовательских задач.

сервер Zabbix

- центральный процесс программного обеспечения Zabbix, выполняющий мониторинг, взаимодействующий с прокси и агентами Zabbix, вычисляющий триггеры и отправляющий уведомления; центральное хранилище данных.

прокси Zabbix

- процесс, который может собирать данные от имени сервера Zabbix, снижая нагрузку на сервер.

агент Zabbix

- процесс, развернутый на объектах мониторинга для активного контроля локальных ресурсов и приложений.

агент Zabbix 2

- агент Zabbix нового поколения для активного контроля локальных ресурсов и приложений, позволяющий использовать пользовательские плагины для мониторинга.

> **Attention:**
> Поскольку агент Zabbix 2 обладает большей частью функциональности агента Zabbix, термин «агент Zabbix» в документации обозначает оба варианта — агент Zabbix и агент Zabbix 2, если их функциональное поведение совпадает. Агент Zabbix 2 упоминается отдельно только в тех случаях, когда его функциональность отличается.

шифрование

- поддержка зашифрованного обмена данными между компонентами Zabbix (сервером, прокси, агентом, утилитами zabbix_sender и zabbix_get) с использованием протокола Transport Layer Security (TLS).

автоматическая регистрация агента

- автоматизированный процесс, в ходе которого агент Zabbix самостоятельно регистрируется как узел сети и начинает мониторинг.

обнаружение сети

- *автоматизированное обнаружение сетевых устройств*.

низкоуровневое обнаружение

- *автоматизированное обнаружение низкоуровневых сущностей на определенном устройстве* (например, файловых систем, сетевых интерфейсов и т. д.).

правило низкоуровневого обнаружения

- *набор определений для автоматизированного обнаружения низкоуровневых сущностей* на устройстве.

прототип элемента данных

- *метрика с определенными параметрами в качестве переменных, готовая к низкоуровневому обнаружению.* После низкоуровневого обнаружения переменные автоматически заменяются фактически обнаруженными параметрами, и метрика автоматически начинает собирать данные.

прототип триггера

- *триггер с определенными параметрами в качестве переменных, готовый к низкоуровневому обнаружению.* После низкоуровневого обнаружения переменные автоматически заменяются фактически обнаруженными параметрами, и триггер автоматически начинает оценивать данные.

Прототипы некоторых других сущностей Zabbix также используются в низкоуровневом обнаружении — прототипы графиков, прототипы узлов сети и прототипы групп узлов сети.

Zabbix Cloud

Zabbix Cloud

Ознакомьтесь с доступными параметрами конфигурации и элементами управления доступом для организаций и узлов в Zabbix Cloud.

Обратите внимание, что узлы Zabbix Cloud в настоящее время работают на **Zabbix 7.0**.
 Подробную информацию о Zabbix 7.0 см. в официальной документации.

Настройка и управление Zabbix Cloud

Начните использовать Zabbix Cloud Пошаговые инструкции, которые помогут быстро запустить ваш экземпляр Zabbix Cloud, включая начальную настройку и подключение контролируемых устройств.

Конфигурация узла Узнайте, как настроить ваш узел Zabbix Cloud, уделяя особое внимание доступу, обработке данных и настройкам ресурсов.

Добавление пользователей Узнайте, как добавлять пользователей и назначать им роли в организации Zabbix Cloud и ее узлах.

Узнайте о Zabbix Cloud

Zabbix Cloud vs. локальное развертывание Сравните различия в управлении, масштабируемости и обслуживании между этими двумя вариантами развертывания и узнайте, какой из них лучше всего поддержит ваш рабочий процесс.

Изучите Zabbix Cloud Получите обзор Zabbix Cloud, включая его ключевые возможности, преимущества и то, что входит в подписку. Ответы на часто задаваемые вопросы по настройке и использованию см. в FAQ.

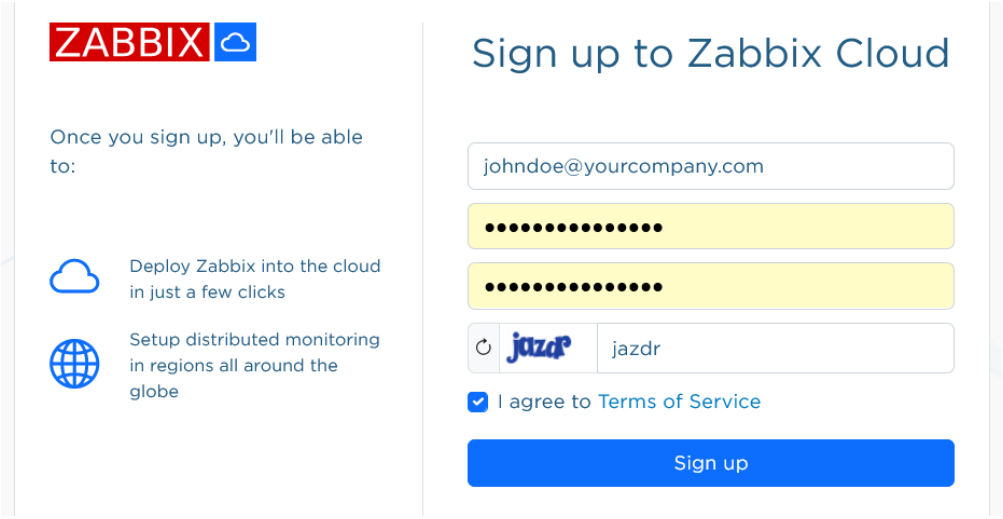
Развертывание Zabbix в облаке

Чтобы начать использовать Zabbix в облаке, выполните следующие действия:

1. Войдите в систему или зарегистрируйтесь на cloud.zabbix.com.
2. Создайте облачный узел сети.
3. Начните использовать Zabbix в облаке.

Вход в систему или регистрация Вы можете войти на cloud.zabbix.com, используя существующую учетную запись Google, GitHub или Microsoft.

Также можно перейти на страницу регистрации, чтобы зарегистрироваться.



После регистрации вы получите электронное письмо с подтверждением создания новой учетной записи.

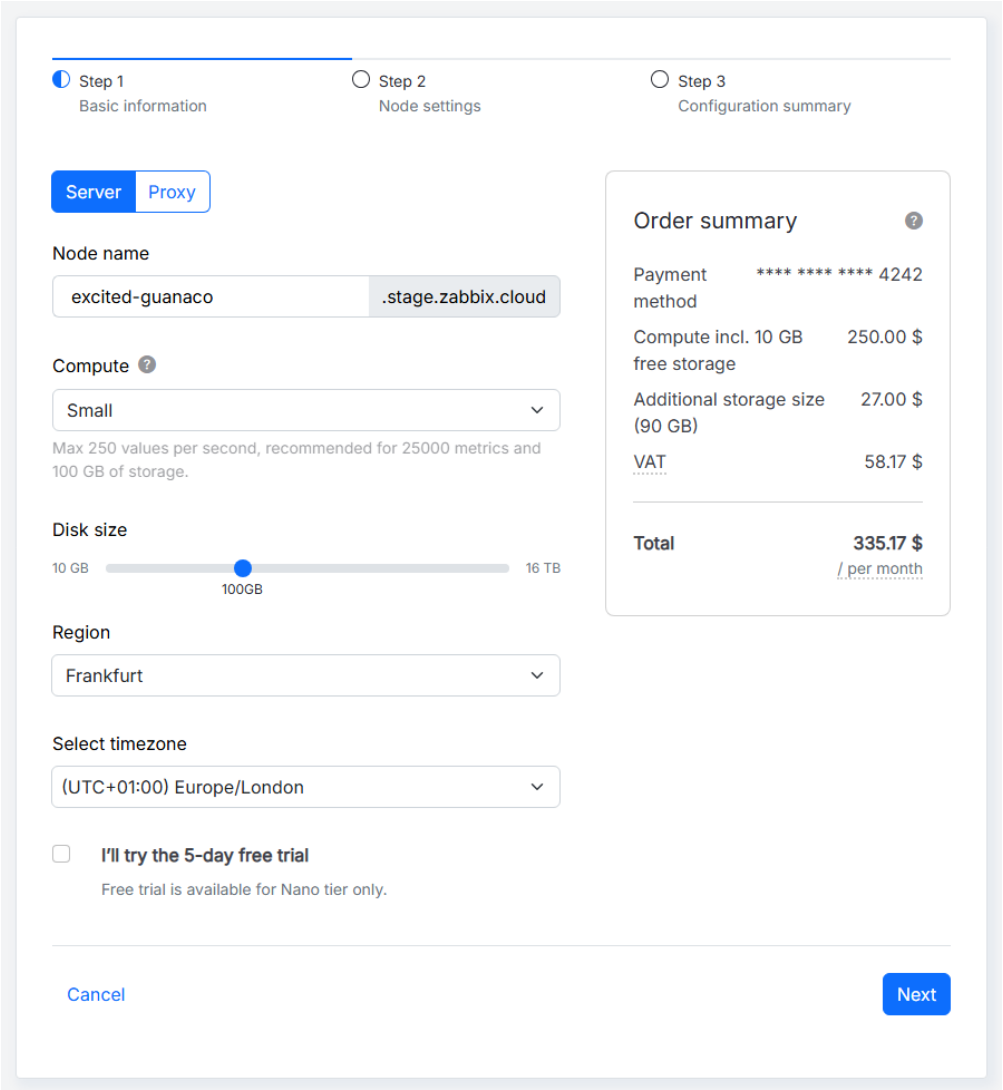
Перейдите на cloud.zabbix.com, чтобы войти в систему. При входе вам будет предложено ввести одноразовый пароль. Этот пароль будет отправлен вам по электронной почте.

Создание облачного узла сети Облачный узел сети — это экземпляр Zabbix, развернутый в Zabbix Cloud. Он может быть развернут как узел сети сервера или узел сети прокси.

После входа в систему вы увидите, что для вашей учетной записи по умолчанию уже существует *Private* организация. Позднее вы сможете переименовать организацию по умолчанию.

Нажмите *Create New Node*. Откроется форма настройки узла сети, состоящая из трех шагов.

На первом шаге необходимо указать основную информацию.



Parameter	Description
Server/Proxy	Выберите, развернуть ли экземпляр сервера Zabbix или экземпляр прокси Zabbix.
Node name	Имя узла сети образует уникальный поддомен в формате `<your-name>.zabbix.cloud`, который используется для доступа к службам узла сети.
Compute	Выберите вычислительную мощность узла сети. Чтобы определить необходимый уровень ресурсов, ознакомьтесь с информацией о ценах. При необходимости вы всегда сможете повысить уровень позднее. Перед созданием узла сети, не являющегося пробным, потребуется указать способ оплаты. Добавьте платежную информацию через меню *Billing*. Для проверки работы узла сети уровня *Nano* доступен **бесплатный пробный период** продолжительностью 5 дней. В течение пробного периода плата не взимается, но после его окончания узел сети будет удален, если его не преобразовать в платный узел сети.
Disk size	По умолчанию ползунок устанавливается на рекомендуемый минимальный размер в зависимости от уровня. Увеличьте размер, если вы храните большие объемы истории или трендов. Обратите внимание, что диски большего размера увеличивают ежемесячную стоимость и что самостоятельно уменьшить размер диска нельзя — для его уменьшения необходимо обратиться в службу поддержки.
Region	Выберите регион, в котором будет развернут узел сети.
Select timezone	Выберите часовой пояс сервера Zabbix. Этот часовой пояс используется сервером Zabbix и не зависит от часовых поясов обслуживания и профиля пользователя.

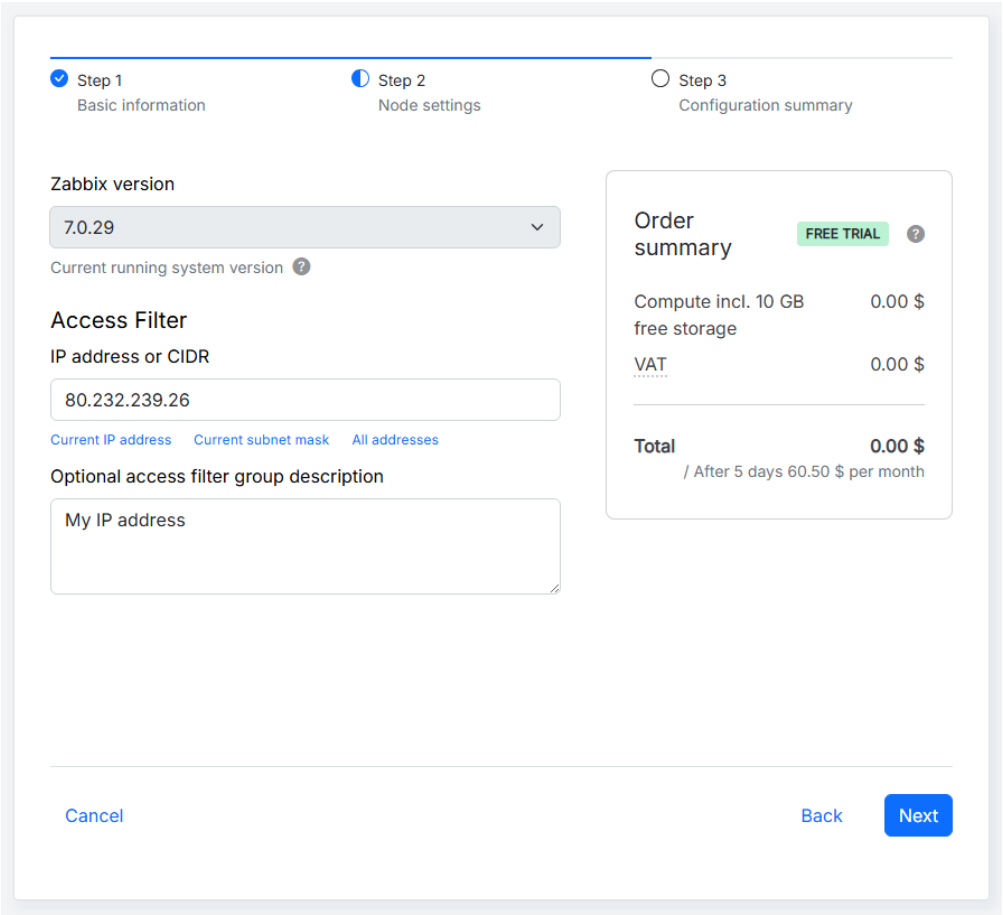
После завершения настройки нажмите *Next*.

На втором шаге необходимо выбрать параметры узла сети.

Параметры настройки на втором шаге зависят от того, создаете ли вы узел сети сервера или прокси.

Настройка узла сети сервера

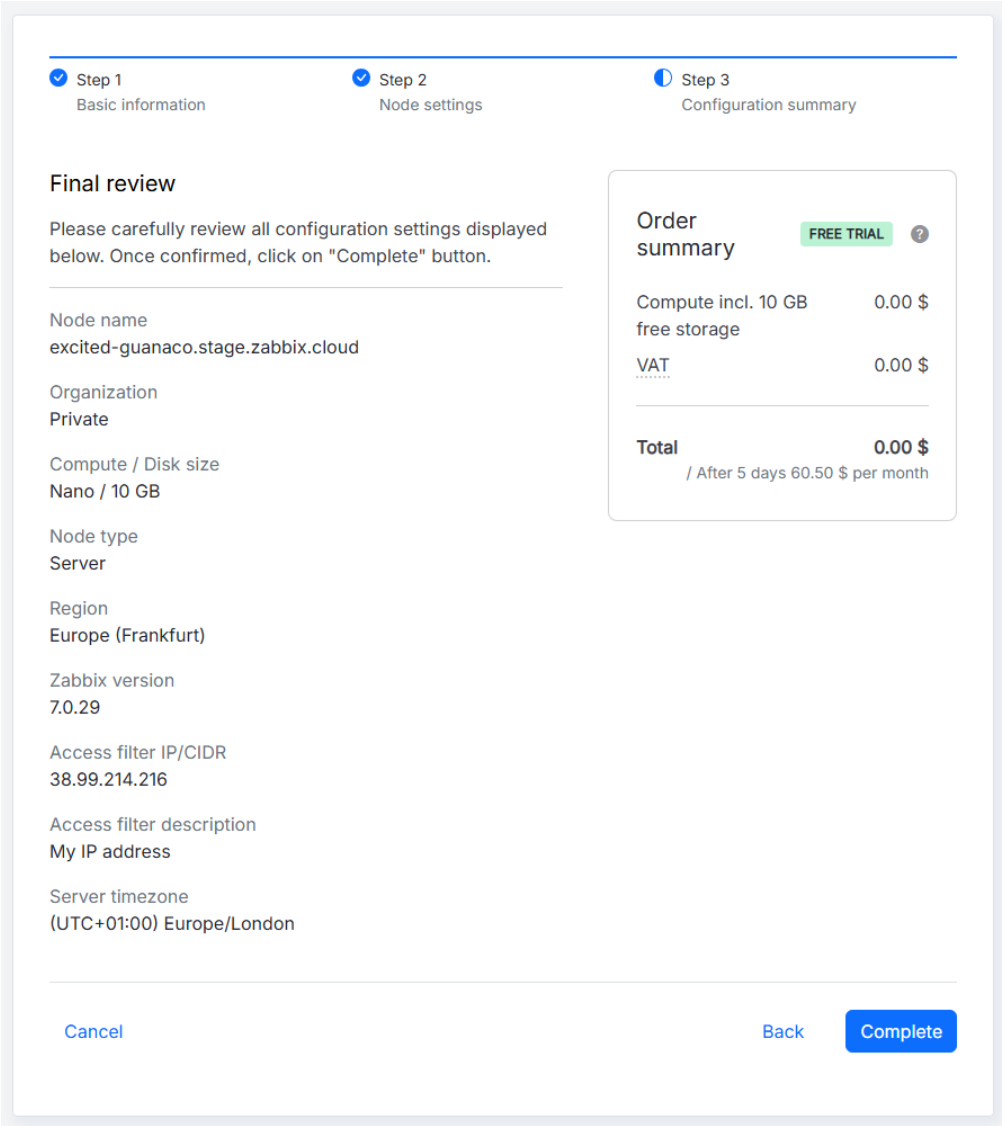
При создании узла сети сервера доступны следующие параметры:



Parameter	Description
Zabbix version	Выберите версию Zabbix. Zabbix Cloud поддерживает последнюю версию LTS и автоматически применяет обновления минорных версий.
IP address or CIDR	Укажите исходный фильтр доступа к узлу сети. Введите IP-адрес или CIDR вручную либо автоматически заполните поле, выбрав соответствующий вариант. Для узлов сети сервера этот фильтр применяется к подключениям к серверу, API и веб-интерфейсу. Позднее фильтр можно изменить на вкладке Access filters.
Optional access filter group description	Добавьте необязательное описание группы фильтров доступа, которая может содержать несколько IP-адресов или CIDR.

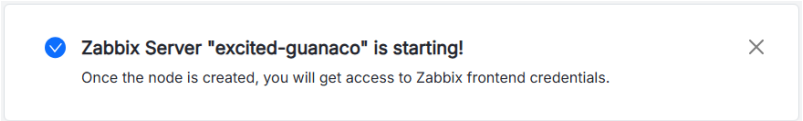
Warning:
При создании узла сети поле *IP address or CIDR* предварительно заполняется IP-адресом, использованным для доступа к Zabbix Cloud. Вы можете изменить это значение, чтобы указать другой исходный фильтр доступа. После создания узла сети вы сможете добавлять, изменять или удалять фильтры доступа на вкладке *Access filters*.

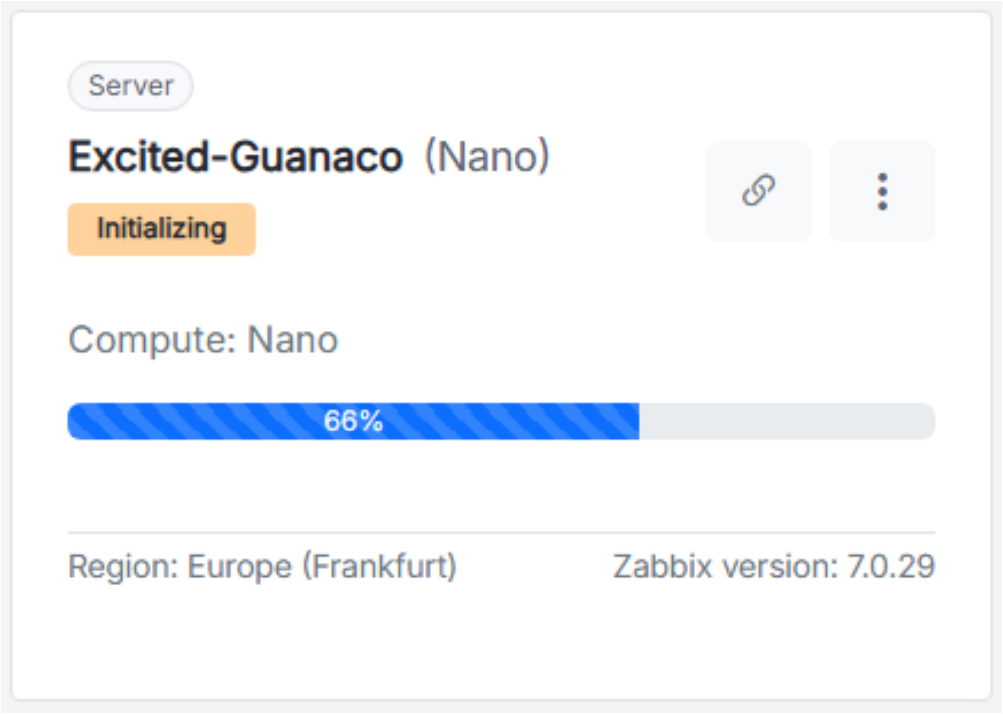
На третьем шаге отображается сводка конфигурации узла сети.



Проверив параметры конфигурации и убедившись, что все указано правильно, нажмите *Complete* для пробного узла сети или *Proceed to payment* для платного узла сети. В противном случае вернитесь к предыдущему шагу, нажав кнопку *Back*, или нажмите *Cancel*, чтобы начать с начала.

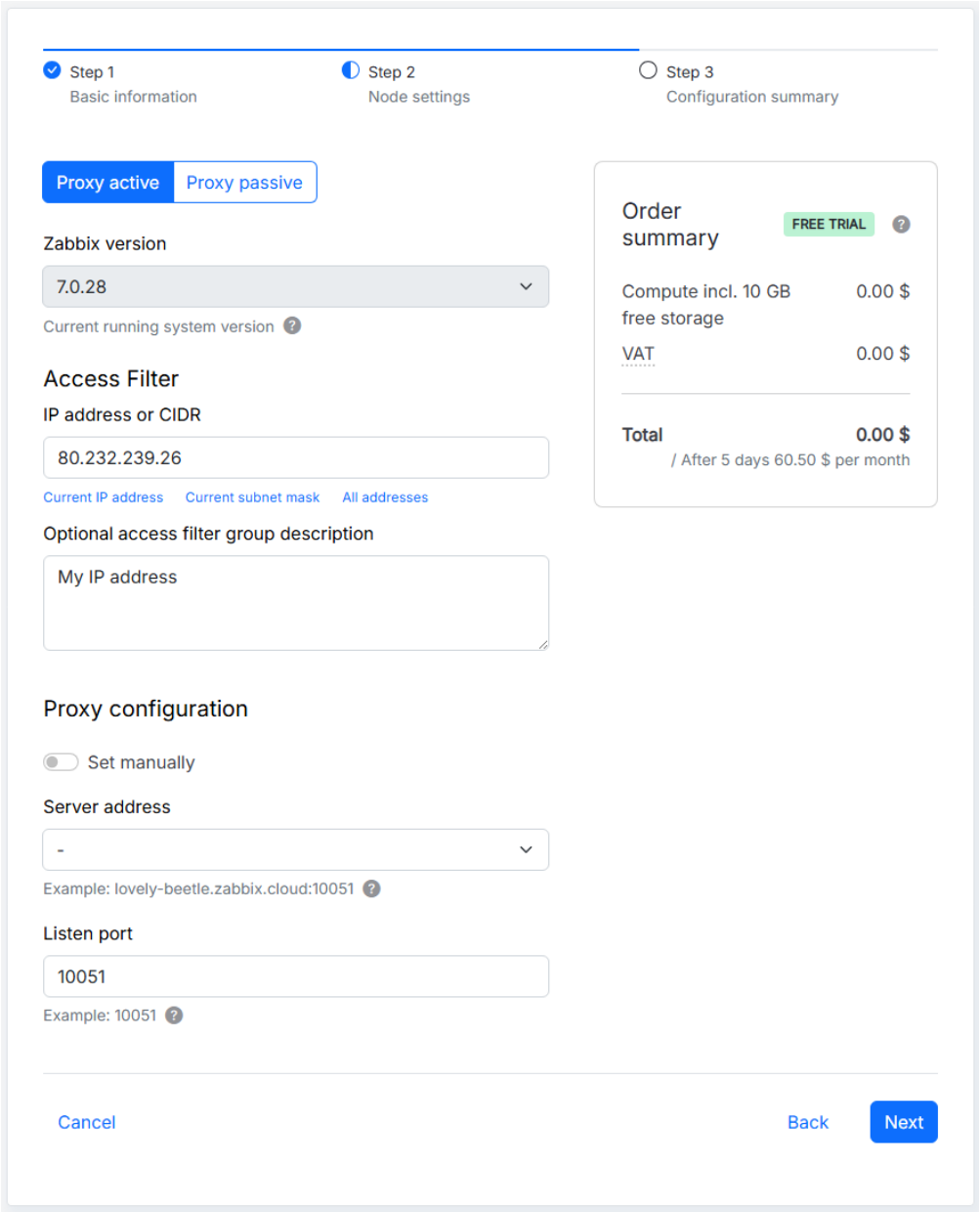
Во время инициализации узла сети отображается соответствующее сообщение, а сам узел сети уже появляется в разделе *Organizations*.





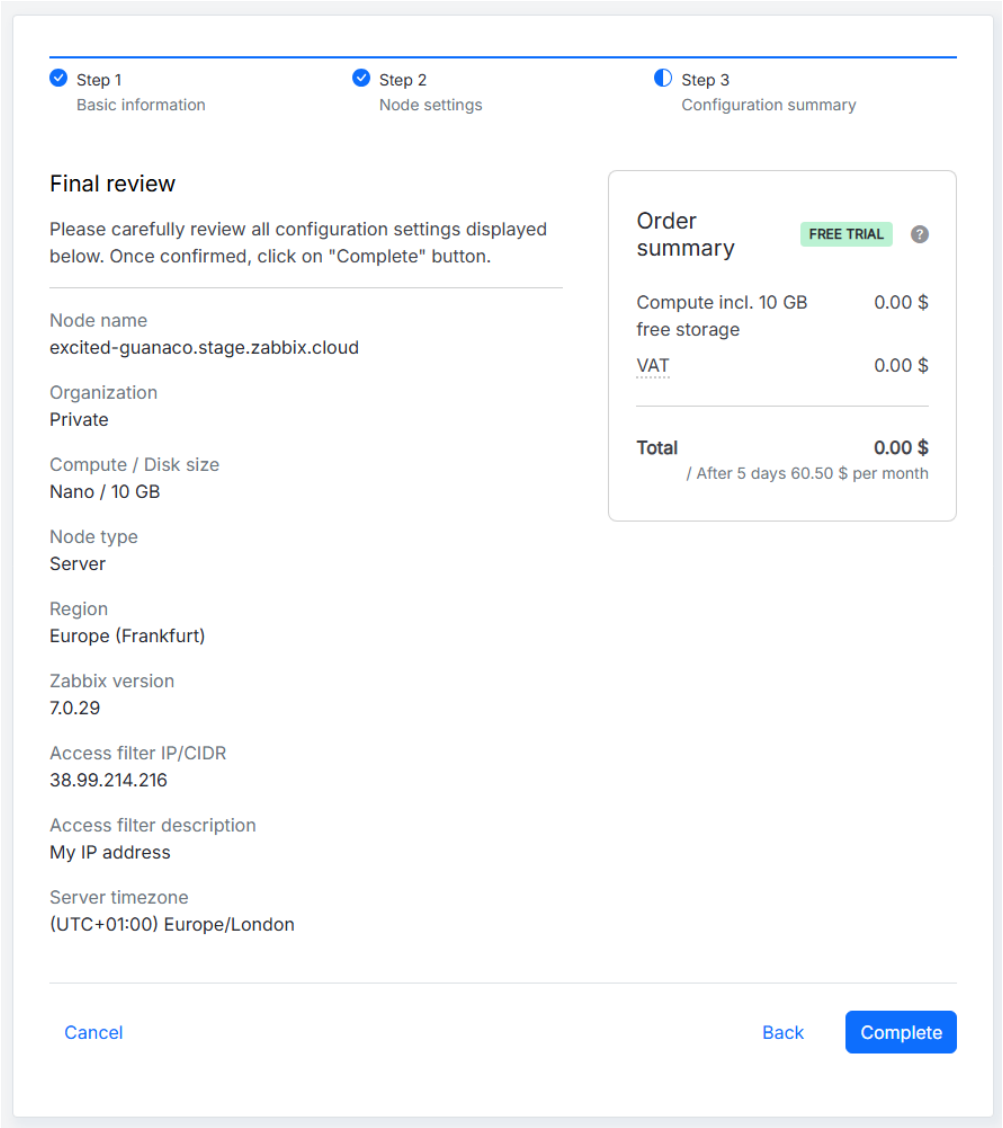
Настройка узла сети прокси

При создании узла сети прокси доступны следующие параметры:



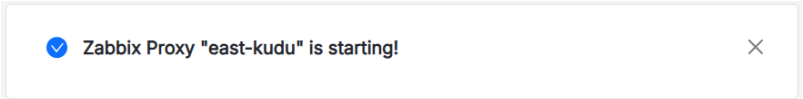
Parameter	Description
Proxy active/Proxy passive	Выберите режим прокси. В активном режиме прокси инициирует подключения к серверу Zabbix; в пассивном режиме сервер Zabbix инициирует подключения к прокси.
Zabbix version	Выберите версию Zabbix. Zabbix Cloud поддерживает последнюю версию LTS и автоматически применяет обновления минорных версий.
IP address or CIDR	Укажите исходный фильтр входящего доступа по IP-адресам для прокси. Введите IP-адрес или CIDR вручную либо автоматически заполните поле, выбрав соответствующий вариант. Фильтр определяет, каким сетевым запросам разрешено достигать прокси, и позднее может быть изменен на вкладке Access filters.
Optional access filter group description	Добавьте необязательное описание группы фильтров доступа, которая может содержать несколько IP-адресов или CIDR.
Set manually	Выберите способ указания сервера Zabbix. Если параметр отключен, выберите узел сети сервера из списка доступных узлов сети серверов. Если параметр включен, укажите адреса серверов вручную.
Server address	Укажите адрес сервера Zabbix. Если параметр *Set manually* отключен, выберите узел сети сервера из списка. Если параметр включен, укажите до 10 адресов серверов. Для активного прокси укажите IP-адрес или DNS-имя сервера Zabbix. Для пассивного прокси укажите IP-адрес, DNS-имя или CIDR сервера Zabbix. Фильтры доступа настраиваются отдельно и не обновляются автоматически при указании адреса сервера.
Listen port	Укажите порт, на котором прокси ожидает входящие подключения. Порт по умолчанию — 10051. Порт прослушивания нельзя изменить после создания узла сети.

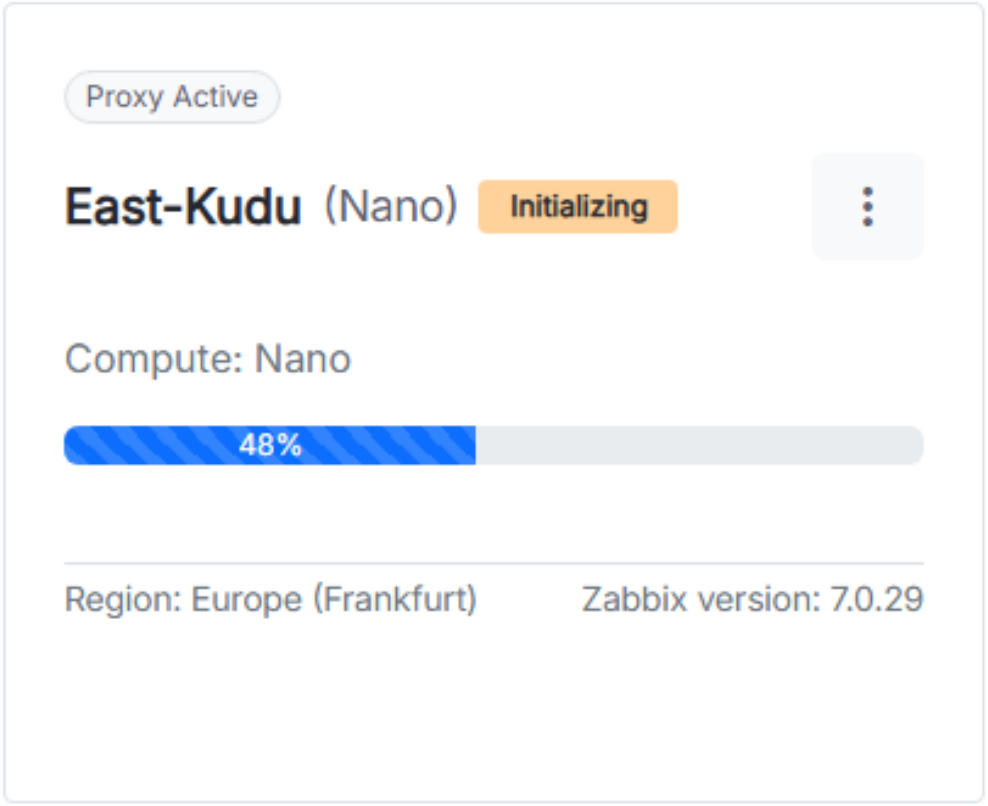
На третьем шаге отображается сводка конфигурации узла сети.



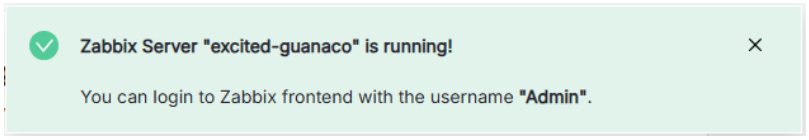
Проверив параметры конфигурации и убедившись, что все указано правильно, нажмите *Complete* для пробного узла сети или *Proceed to payment* для платного узла сети. В противном случае вернитесь к предыдущим шагам, нажав кнопку *Back*, или нажмите *Cancel*, чтобы начать с начала.

Во время инициализации узла сети отображается соответствующее сообщение, а сам узел сети уже появляется в разделе *Organizations*.





Информацию о настройке прокси Zabbix Cloud см. в разделе Настройка прокси Zabbix Cloud.

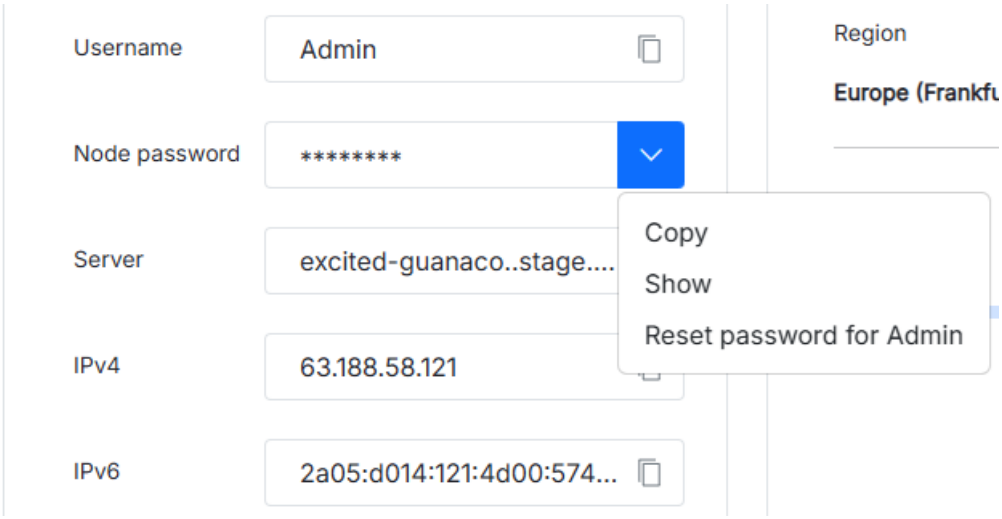


Начало работы с Zabbix

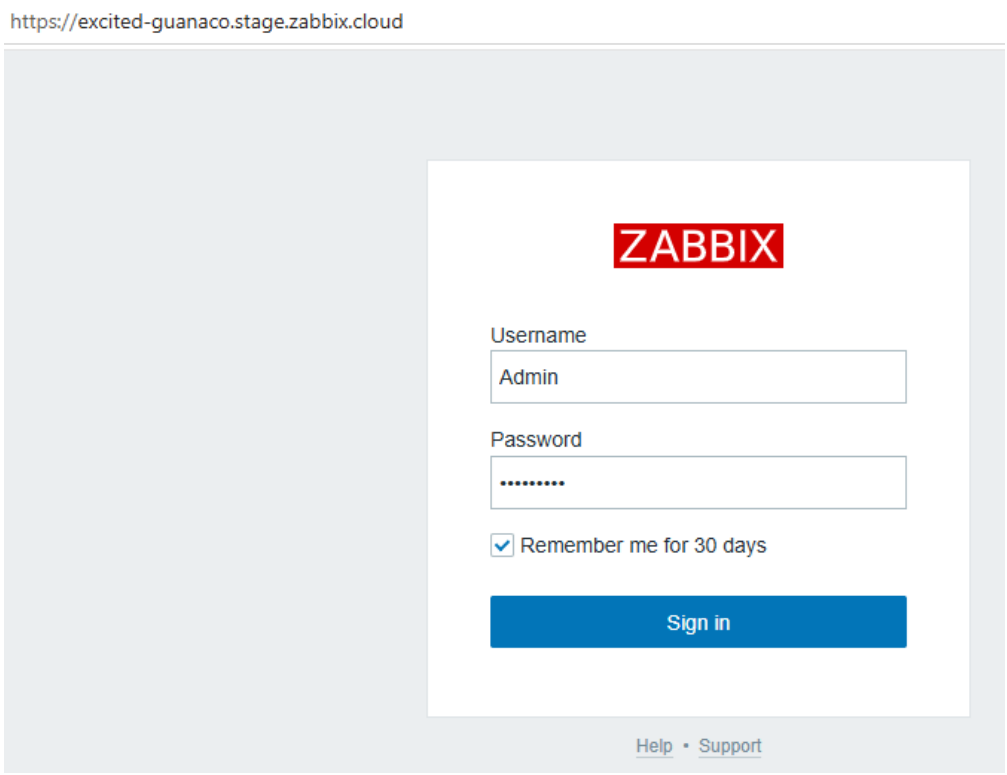
Когда статус узла сети сервера изменится с *Initializing* на *Running*, вы сможете войти в веб-интерфейс Zabbix.

Чтобы получить первоначальный пароль веб-интерфейса, нажмите на созданный узел сети.

Вы перейдёте к настройкам узла сети (обзор, фильтры доступа, шифрование, резервные копии и т. д.). В разделе **Connection info** найдите поле *Node password* и выберите в раскрывающемся списке пункт *Reset password for Admin*. После сброса пароля снова нажмите на этот раскрывающийся список и выберите *Copy*.



Нажмите URL веб-интерфейса, чтобы открыть веб-интерфейс Zabbix (`excited-guanaco.zabbix.cloud`).



Войдите, используя следующие учётные данные:

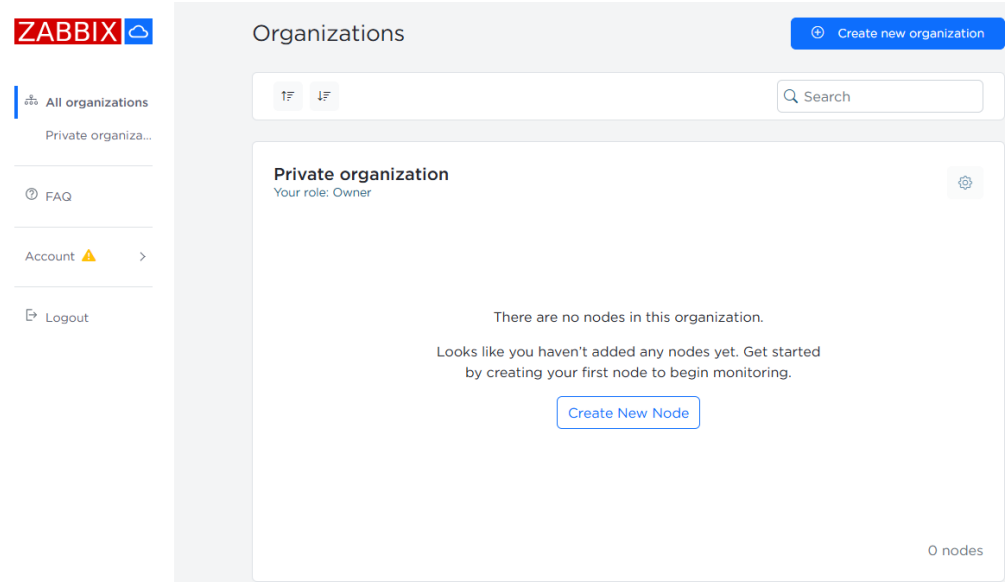
- *Username* - `Admin`
- *Password* - укажите скопированный пароль

Чтобы начать мониторинг, необходимо определить фильтры доступа - IP-адрес или IP-адреса, которым разрешено подключаться к серверу Zabbix Cloud. Необходимо определить хотя бы один фильтр.

Управление организациями Организация — это контейнер для одного или нескольких узлов сети. Организации также помогают структурировать ресурсы и делегировать обязанности, например:

- Управление узлами сети
- Выставление счетов

При входе в Zabbix Cloud отображается страница *Все организации*. По умолчанию для вашей учетной записи создается частная организация.

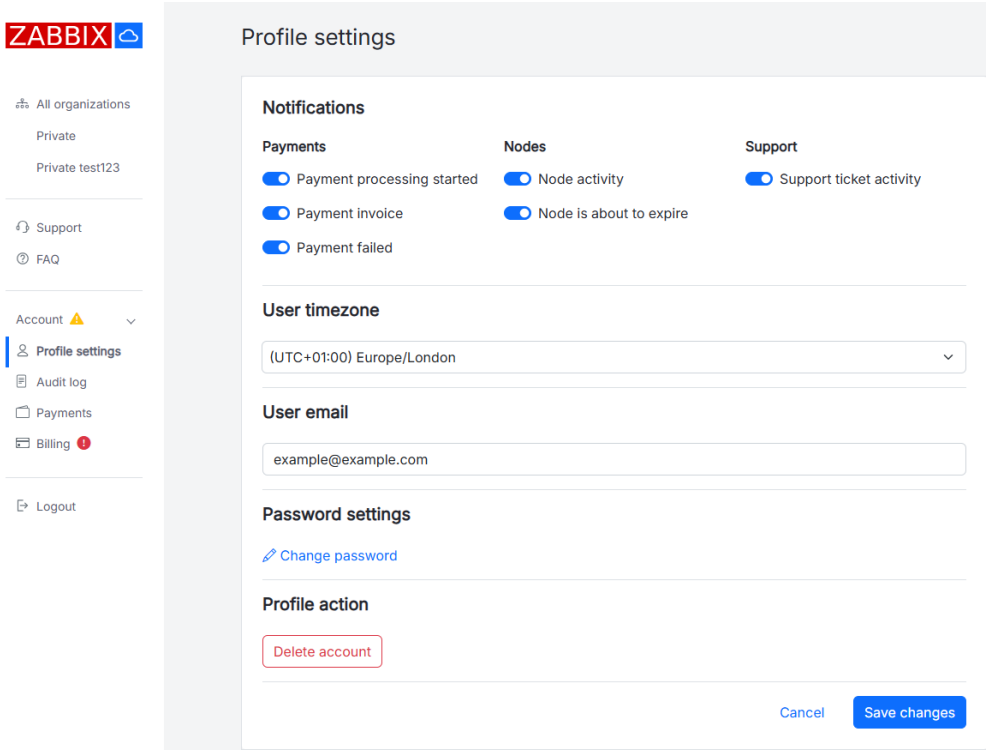


Как владелец этой организации, вы можете управлять пользователями, администраторами, узлами сети и названием организации. Нажмите кнопку настроек , чтобы получить доступ к параметрам управления.

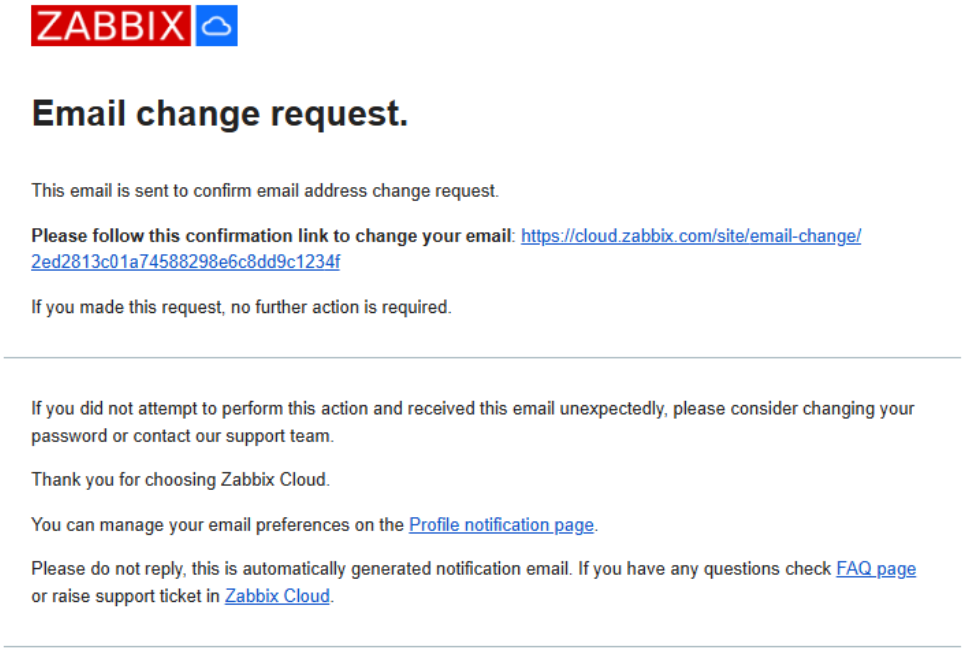
Управление учетными записями Изменение адреса электронной почты учетной записи

Чтобы изменить адрес электронной почты учетной записи, перейдите в раздел *Учетная запись* > *Настройки профиля*.

В поле *Электронная почта пользователя* введите новый адрес электронной почты и нажмите *Сохранить изменения*.



На новый адрес электронной почты будет отправлено сообщение с подтверждением. Перейдите по ссылке в сообщении, чтобы подтвердить изменение адреса электронной почты.



По ссылке для подтверждения откроется страница *Настройки профиля* с сообщением о подтверждении и обновленным адресом электронной почты в поле «Электронная почта пользователя».

После подтверждения изменения на новый и предыдущий адреса электронной почты будет отправлено сообщение с подтверждением изменения. В сообщении, отправленном на предыдущий адрес, указаны предыдущий и новый адреса электронной почты.

Конфигурация узла сети

Обзор

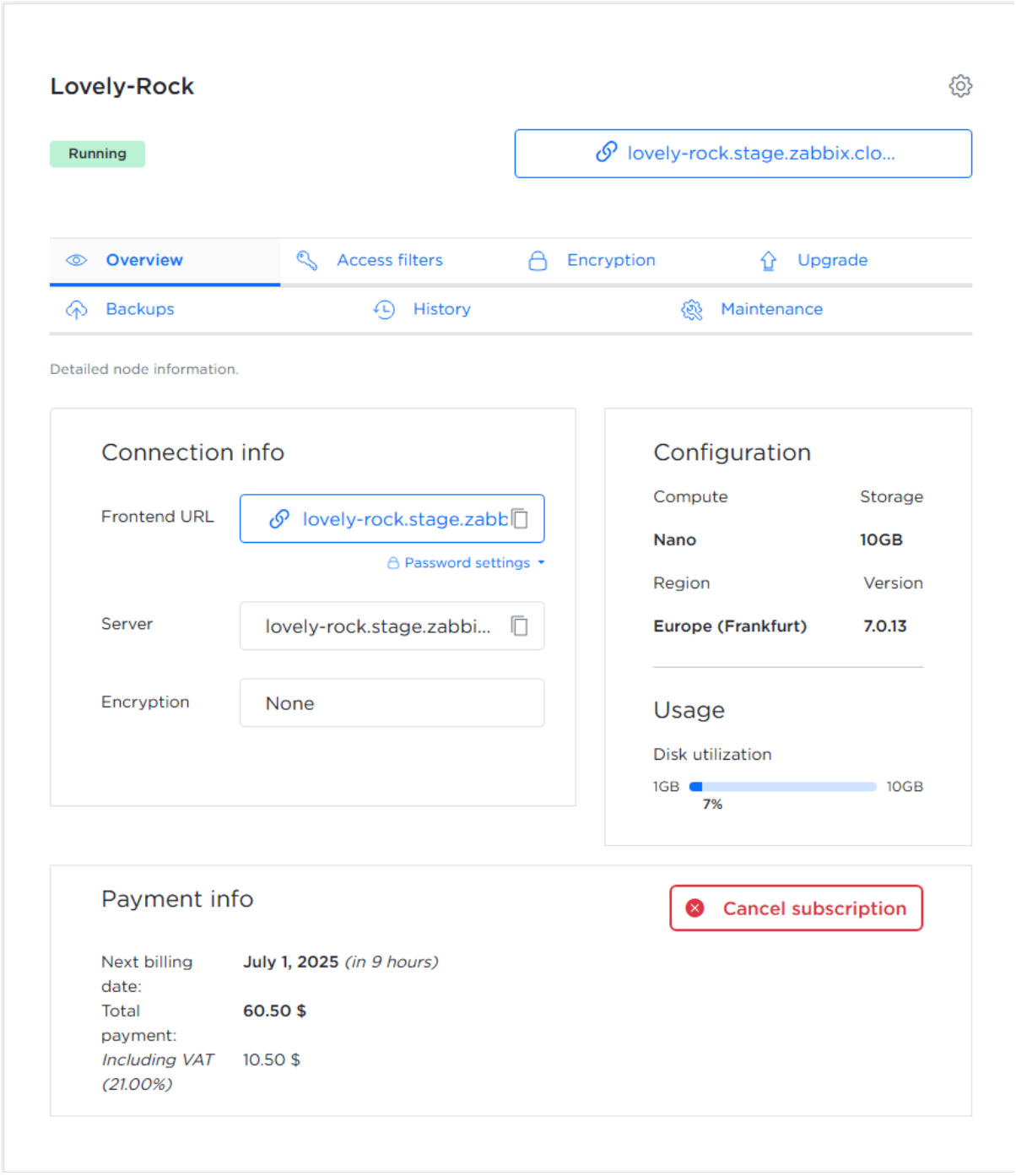
Облачный узел — это компонент Zabbix, развернутый в облаке.
Узлы сервера включают сервер Zabbix, базу данных и веб-интерфейс, а узлы прокси — прокси Zabbix и базу данных.
Облачный узел имеет выделенные вычислительные ресурсы (ЦП, память) и дисковое хранилище.

Пользователи могут создавать один или несколько облачных узлов.
Облачные узлы можно объединять в **организации** (структурные, региональные и т. д.), создавая распределенную систему мониторинга.

Облачные узлы и организации создаются с помощью пользовательского интерфейса Zabbix Cloud.
Чтобы открыть страницу настройки узла, нажмите на узел в разделе *Все организации*, в меню организации или перейдите по ссылке *Страница настройки узла*, отображаемой после запуска узла.

В представлении настройки узла каждая вкладка позволяет управлять определенным аспектом узла:

- Обзор — просмотр сведений об узле, открытие его веб-интерфейса и управление пользователями узла.
- Фильтры доступа — управление доступом по IP-адресу или подсети.
- Шифрование — настройка TLS путем загрузки сертификатов, ключей и необязательных списков отзыва сертификатов (CRL).
- Резервные копии — создание, восстановление и удаление снимков узла; просмотр времени следующего запланированного резервного копирования.
- История — настройка сроков хранения журналов аудита, истории и трендов с использованием предустановленных единиц измерения и ограничением обновления до 3 раз в день.
- Обслуживание — определение еженедельных периодов обслуживания с указанием часового пояса, дня и часа.
- Обновление — изменение вычислительных ресурсов и ресурсов хранилища либо управление статусом пробного периода и оплаты.

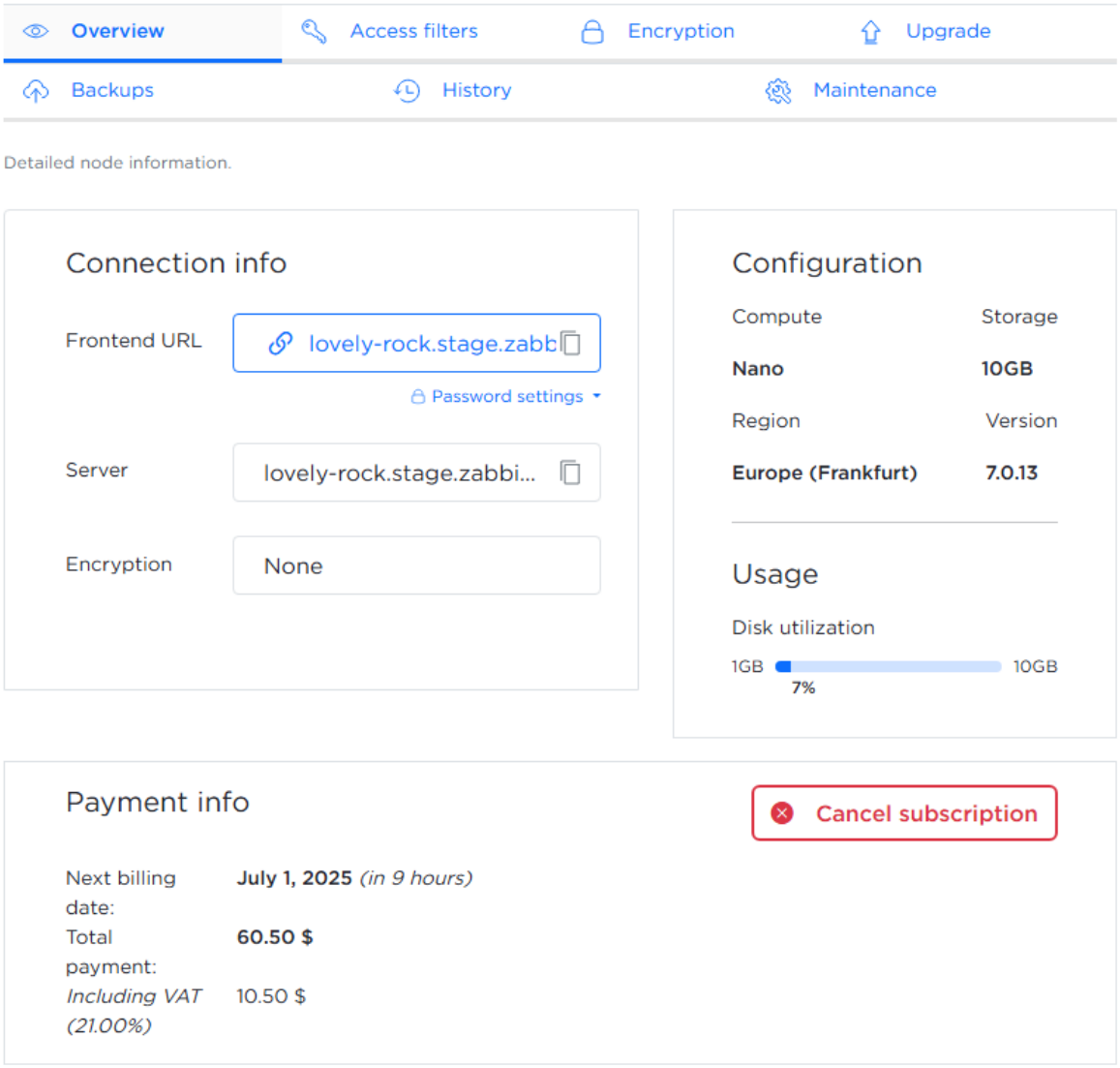


Note:
Некоторые вкладки могут быть недоступны в зависимости от типа узла (сервер или прокси), типа пользователя, просматривающего страницу, и статуса узла (пробный период или не пробный период).

Вкладка «Обзор»

На вкладке *Обзор* представлена подробная информация об узле сети. Отображаемая информация зависит от того, является ли узел сети сервером или прокси.

Узел сервера Вкладка *Обзор* для узла сервера содержит следующие разделы:



Сведения о подключении:

- URL веб-интерфейса, доступный для нажатия, с возможностью копирования в буфер обмена
- Имя пользователя
- Пароль узла сети и его настройки
- Имя узла сети сервера с возможностью копирования в буфер обмена
- IPv4-адрес с возможностью копирования в буфер обмена
- IPv6-адрес с возможностью копирования в буфер обмена
- Текущий метод шифрования

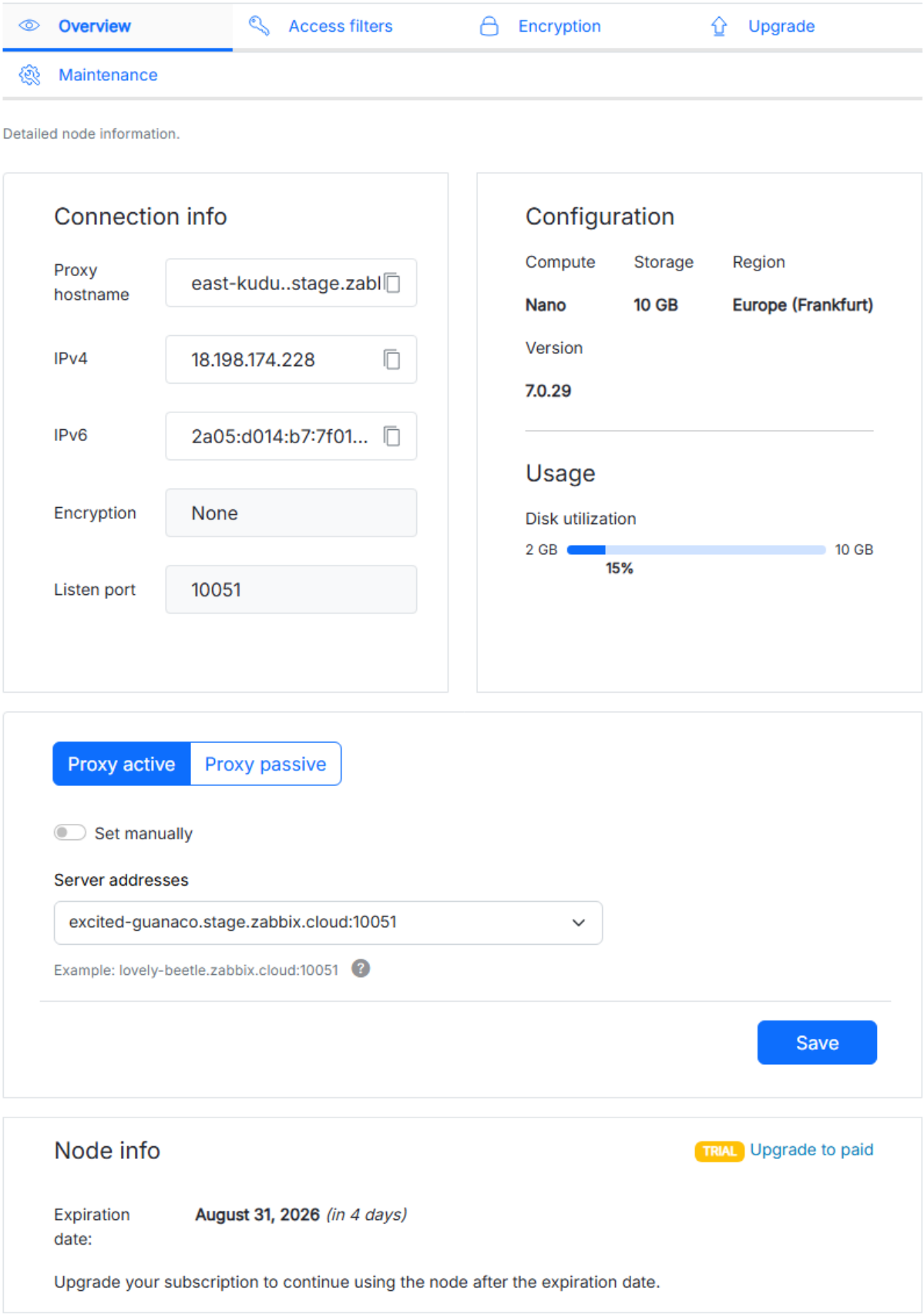
Конфигурация и использование:

- Вычислительный уровень, максимальный объем хранилища, регион
- Версия компонентов Zabbix
- Использование диска

Сведения об узле сети/Сведения об оплате:

- Дата окончания срока действия (для узлов сети с пробным периодом и предоплатой)
- После завершения инициализации узла сети отображается контекстная кнопка в зависимости от типа узла сети: - *Перейти на платный план* - доступно для узлов сети с бесплатным пробным периодом - *Отменить подписку* - доступно для платных узлов сети с активной подпиской - *Повторно оформить платную подписку* - отображается, если подписка была отменена - Для узлов сети с предоплатой кнопка действия не отображается: они имеют длительный срок действия и не могут быть отменены

Прокси-узел Вкладка *Обзор* для прокси-узла содержит следующие разделы:



Информация о подключении:

- Имя узла сети прокси с возможностью копирования в буфер обмена
- IPv4-адрес с возможностью копирования в буфер обмена
- IPv6-адрес с возможностью копирования в буфер обмена
- Текущий метод шифрования
- Текущий порт прослушивания

Конфигурация и использование:

- Уровень вычислительных ресурсов, максимальный объем хранилища, регион
- Версия компонентов Zabbix
- Использование диска

Режим прокси:

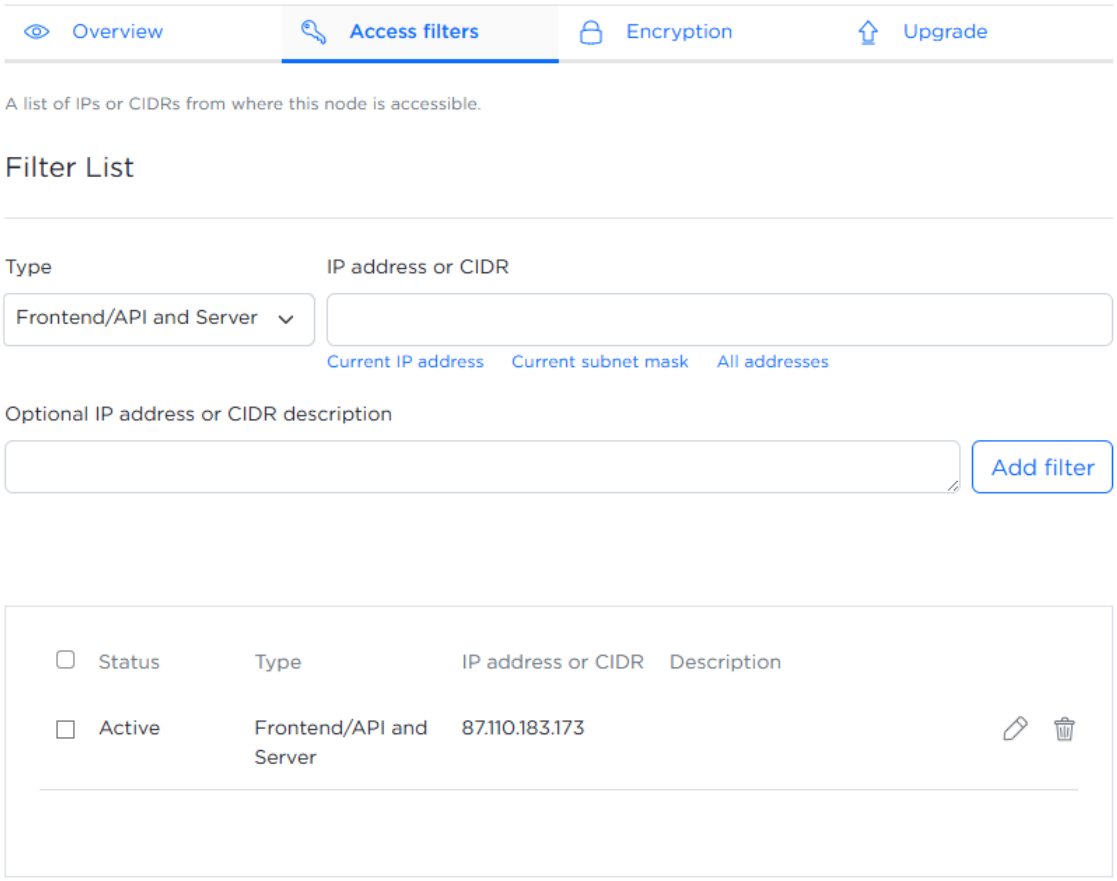
- Переключение между активным и пассивным режимами прокси
- Выбор адреса сервера в раскрывающемся списке *Адреса серверов* или включение параметра *Задать вручную* для ручного ввода адреса сервера

Информация об узле сети/Информация об оплате:

- Дата окончания срока действия (для узлов сети с пробным периодом и предоплатой)
- После завершения инициализации узла сети отображается контекстная кнопка в зависимости от типа узла сети: - *Перейти на платный тариф* - доступно для узлов сети с бесплатным пробным периодом - *Отменить подписку* - доступно для платных узлов сети с активной подпиской - *Возобновить платную подписку* - отображается, если подписка была отменена - Для узлов сети с предоплатой кнопка действия не отображается, поскольку они имеют длительный срок действия и не могут быть отменены

Вкладка «Фильтры доступа»

Вкладка *Фильтры доступа* позволяет ограничить доступ к вашему узлу сети Zabbix по IP-адресу или подсети.



Чтобы настроить фильтр доступа, заполните следующие поля:

- *Тип* - для узлов сети сервера с помощью раскрывающегося списка выберите компонент Zabbix, доступ к которому необходимо ограничить: только веб-интерфейс/API, только сервер или оба компонента. Для узлов сети прокси доступен только параметр *Входящий IP-адрес*. Он определяет, какие входящие сетевые запросы разрешено направлять к прокси.
- *IP-адрес или CIDR* - введите IP-адрес или CIDR вручную либо заполните поле автоматически, выбрав соответствующий вариант.
- *Необязательное описание IP-адреса или CIDR* - добавьте необязательное описание для каждого IP-адреса или подсети.

После заполнения полей нажмите *Добавить фильтр*. Добавленные фильтры отображаются в таблице ниже с указанием состояния, типа, IP-адреса или подсети и описания.

Warning:
Во время инициализации узла сети автоматически создается фильтр доступа для использованного вами IP-адреса. Если позднее вы подключитесь с другого IP-адреса или из другой сети, необходимо добавить этот адрес здесь, иначе доступ будет заблокирован.

Вы можете:

-  - изменить фильтр.
-  - удалить фильтр или несколько фильтров, установив соответствующие флажки.

Вкладка «Шифрование»

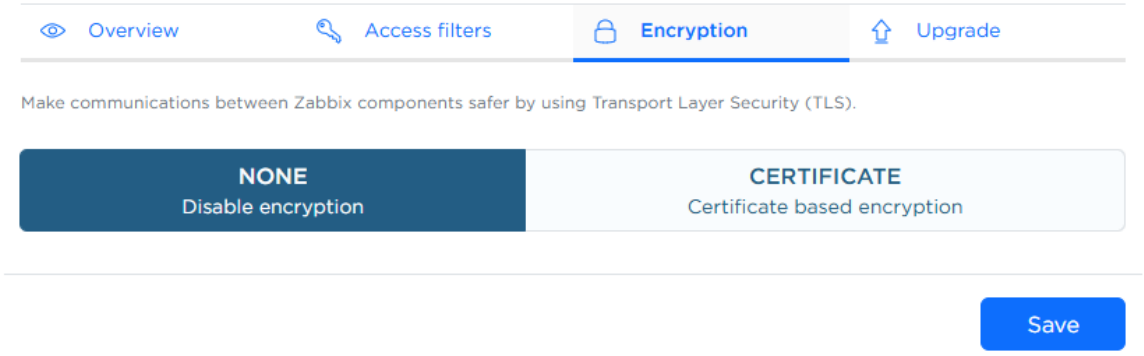
На вкладке *Шифрование* можно настроить шифрование TLS для обмена данными между компонентами Zabbix, загрузив необходимые файлы сертификатов.

Чтобы включить защищенный обмен данными с использованием сертификатов, выберите *СЕРТИФИКАТ* и загрузите со своего компьютера следующие файлы:

- *Корневые сертификаты центра сертификации* - доверенные сертификаты центра сертификации, используемые для проверки сертификатов узлов сети. Нажмите *Выбрать файл* и выберите файл, содержащий доверенные корневые сертификаты центра сертификации (обычно файл `.crt` или `.pem` в кодировке PEM).
- *Цепочка сертификатов* - представляет цепочку доверия для собственного сертификата. Нажмите *Выбрать файл* и выберите файл в кодировке PEM (например, `.crt` или `.pem`), содержащий ваш сертификат и все промежуточные сертификаты центра сертификации.

- *Закрытый ключ* - закрытый ключ, соответствующий вашему сертификату. Нажмите *Выбрать файл* и выберите файл `.key` в кодировке PEM.
- *Отозванные сертификаты* (необязательно) - если вы используете список отозванных сертификатов (CRL), нажмите *Выбрать файл* и выберите файл CRL, чтобы указать сертификаты, которым больше не следует доверять.

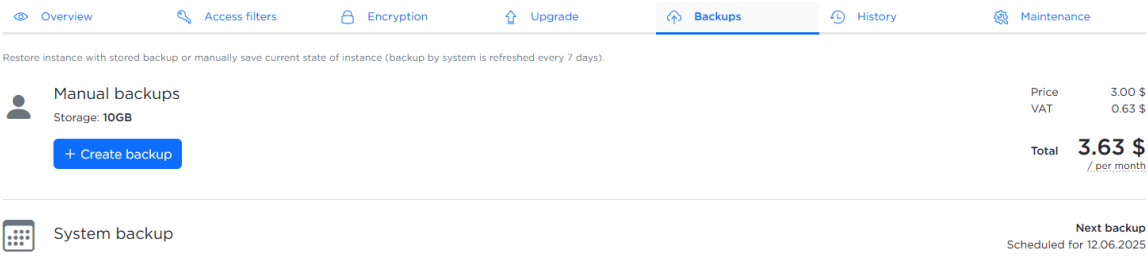
После выбора файлов нажмите *Сохранить*. Файлы будут загружены, проверены и применены для настройки шифрования TLS между компонентами Zabbix.



Вкладка «Резервные копии»

На вкладке *Резервные копии* можно просматривать, создавать, восстанавливать и удалять резервные копии вашего узла сети Zabbix.

Вы можете восстановить свой экземпляр из любой сохраненной резервной копии или в любое время создать новую резервную копию вручную. На вкладке также отображается дата следующего запланированного резервного копирования.



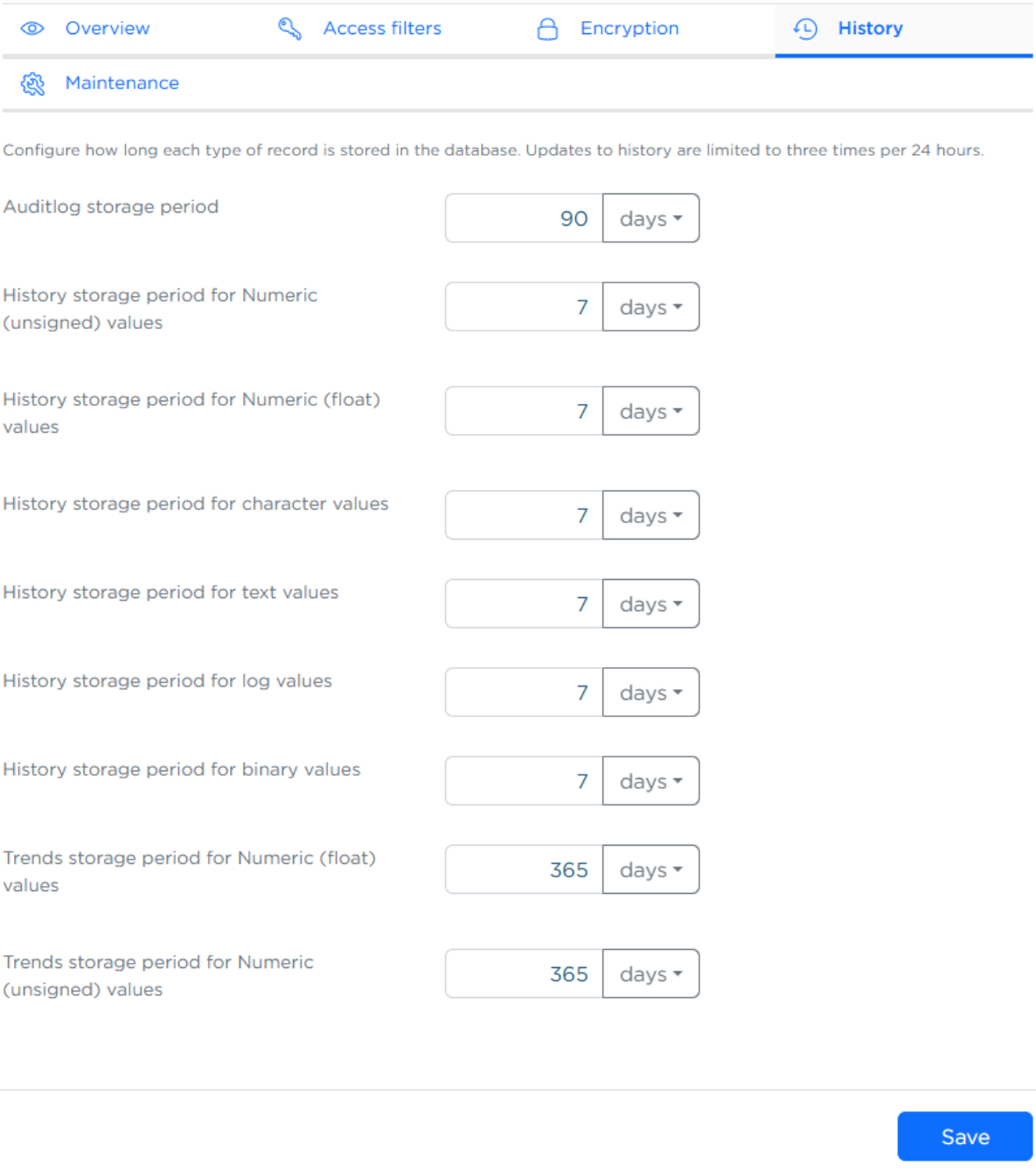
Чтобы создать резервную копию:

1. Нажмите кнопку *Создать резервную копию*.
2. Введите необязательное описание во всплывающем окне.
3. Нажмите *Создать*, чтобы подтвердить действие.

Warning:
За дополнительные резервные копии ежемесячно взимается плата в зависимости от объема используемого хранилища.

Вкладка «История»

На вкладке *История* можно настроить срок хранения различных типов исторических данных.



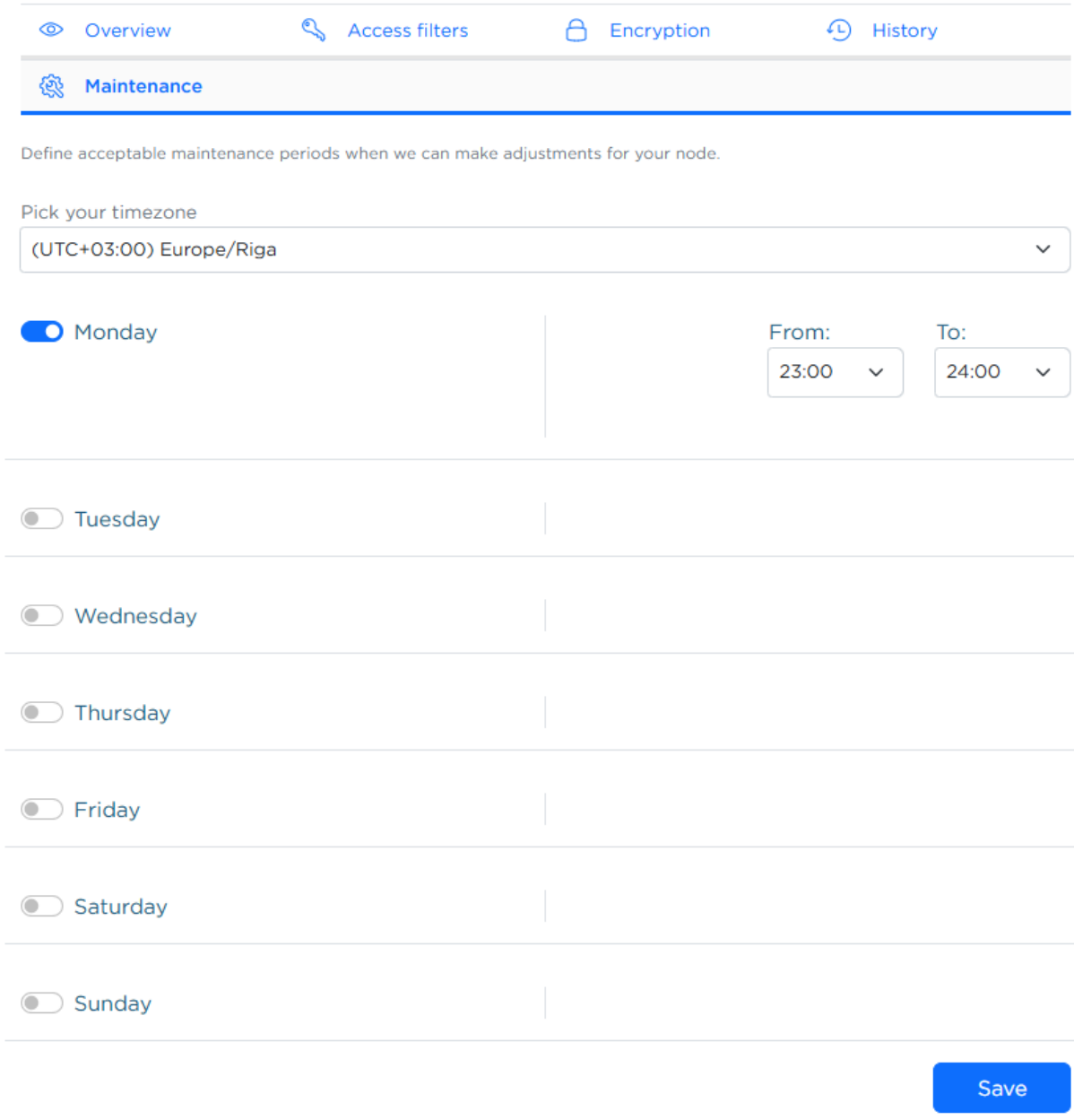
Для каждого типа данных (например, числовых, текстовых данных и журналов) можно задать срок хранения и выбрать единицу измерения (дни, месяцы или годы). Изменять настройки можно не более трех раз в течение 24 часов.

После обновления значений нажмите *Сохранить*, чтобы применить изменения.

Warning:
Более длительные сроки хранения могут увеличить использование дискового пространства и повлиять на производительность. Рекомендуется настраивать сроки хранения с учетом фактического объема данных и потребностей в их анализе.
 Zabbix Cloud использует секционирование базы данных для управления хранением данных вместо традиционной очистки данных Zabbix. Это обеспечивает более высокую производительность и более предсказуемое управление сроками хранения.

Вкладка «Обслуживание»

На вкладке *Обслуживание* можно задать допустимые окна обслуживания, в течение которых на вашем узле сети можно выполнять обновления или настройки.

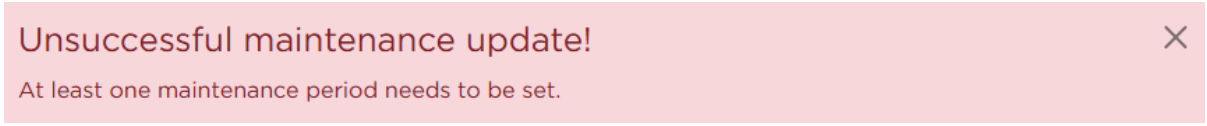


Необходимо:

- Выбрать часовой пояс в раскрывающемся списке.
- Выбрать как минимум один день и одно почасовое окно для обслуживания.

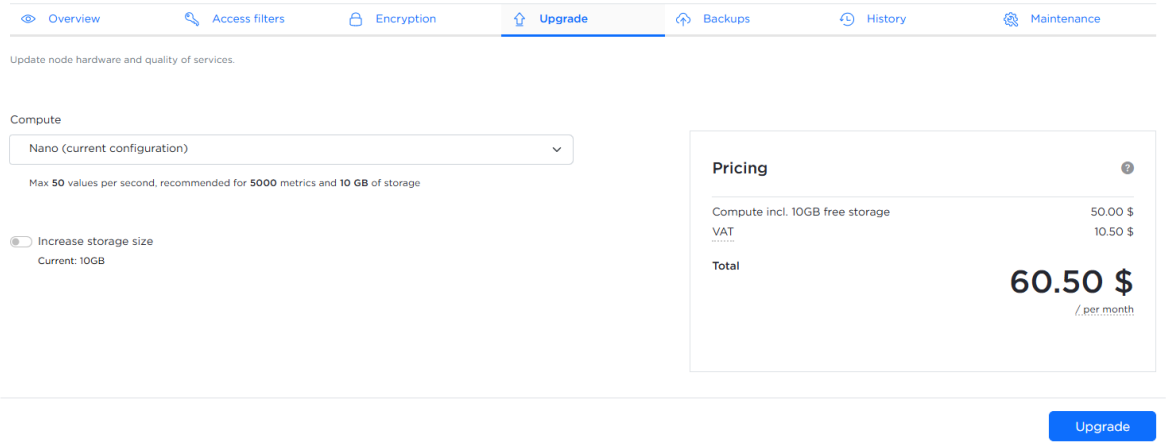
Нажмите *Сохранить*, чтобы применить изменения.

Если день или временное окно не выбраны, будет отображено сообщение об ошибке.



Вкладка «Обновление»

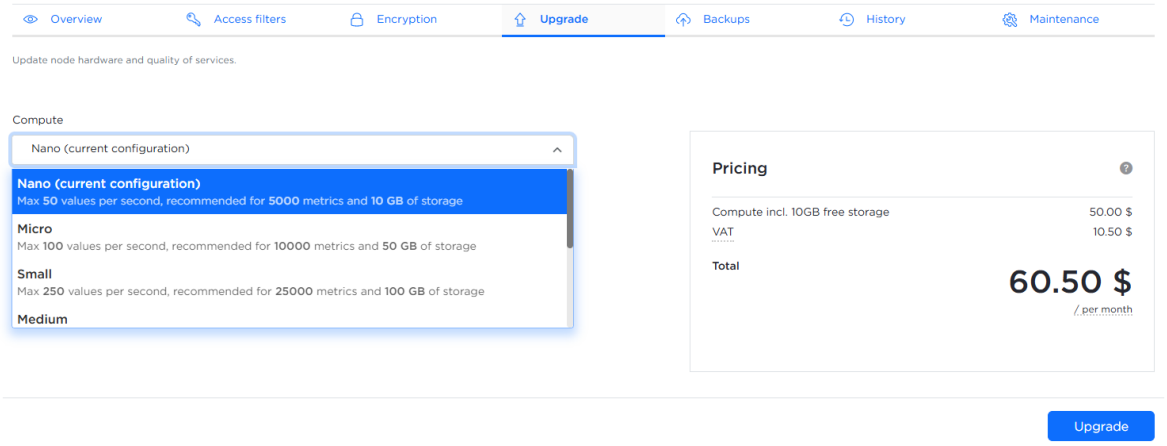
Вкладка *Обновление* позволяет увеличить ресурсы узла сети или управлять его подпиской.



Вы можете увеличить вычислительный уровень или размер диска. Обратите внимание, что размер диска можно увеличивать только один раз в 24 часа, а уменьшить его через веб-интерфейс нельзя (для запроса на уменьшение размера обратитесь в службу поддержки Zabbix). Узлы сети с бесплатной пробной подпиской можно перевести на платные подписки.

Чтобы увеличить вычислительный уровень:

1. Нажмите раскрывающийся список *Вычислительные ресурсы*.



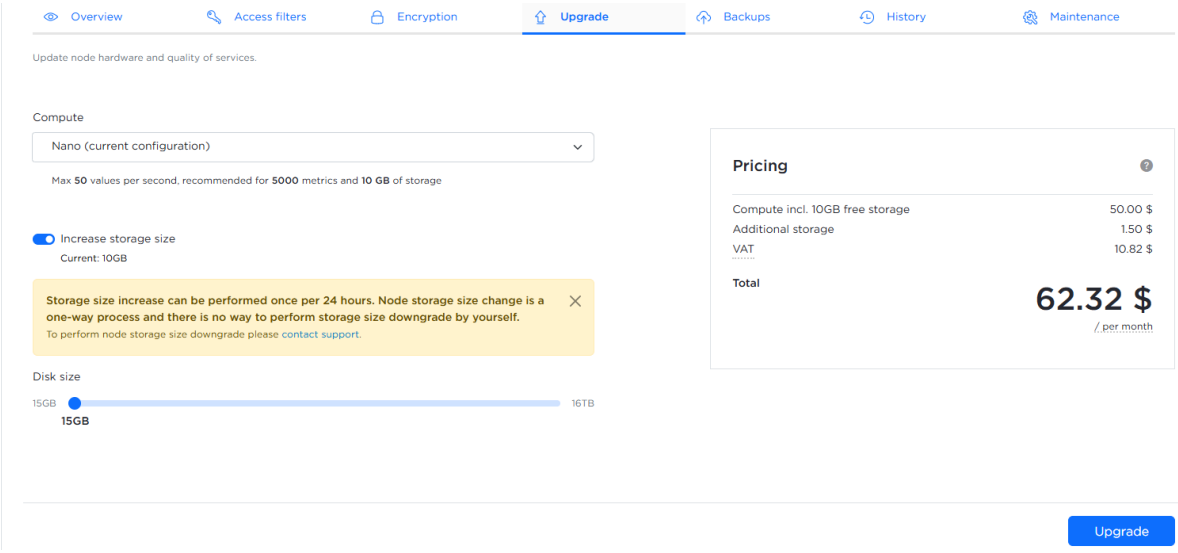
2. Выберите нужный уровень конфигурации.

3. Нажмите кнопку *Обновить*.

Чтобы увеличить размер диска:

1. Выберите переключатель *Увеличить размер хранилища*.

2. Установите ползунок *Размер диска* на нужное значение.



3. Нажмите *Обновить*.

Note:
Обновленная общая ежемесячная стоимость отображается в таблице *Цены* перед подтверждением изменений.

Добавление пользователей

Введение

Следующие примеры описывают, как добавить пользователей в вашу организацию и узлы.

Note:
Копирование, очистка и сброс пароля веб-интерфейса могут выполняться только владельцем.

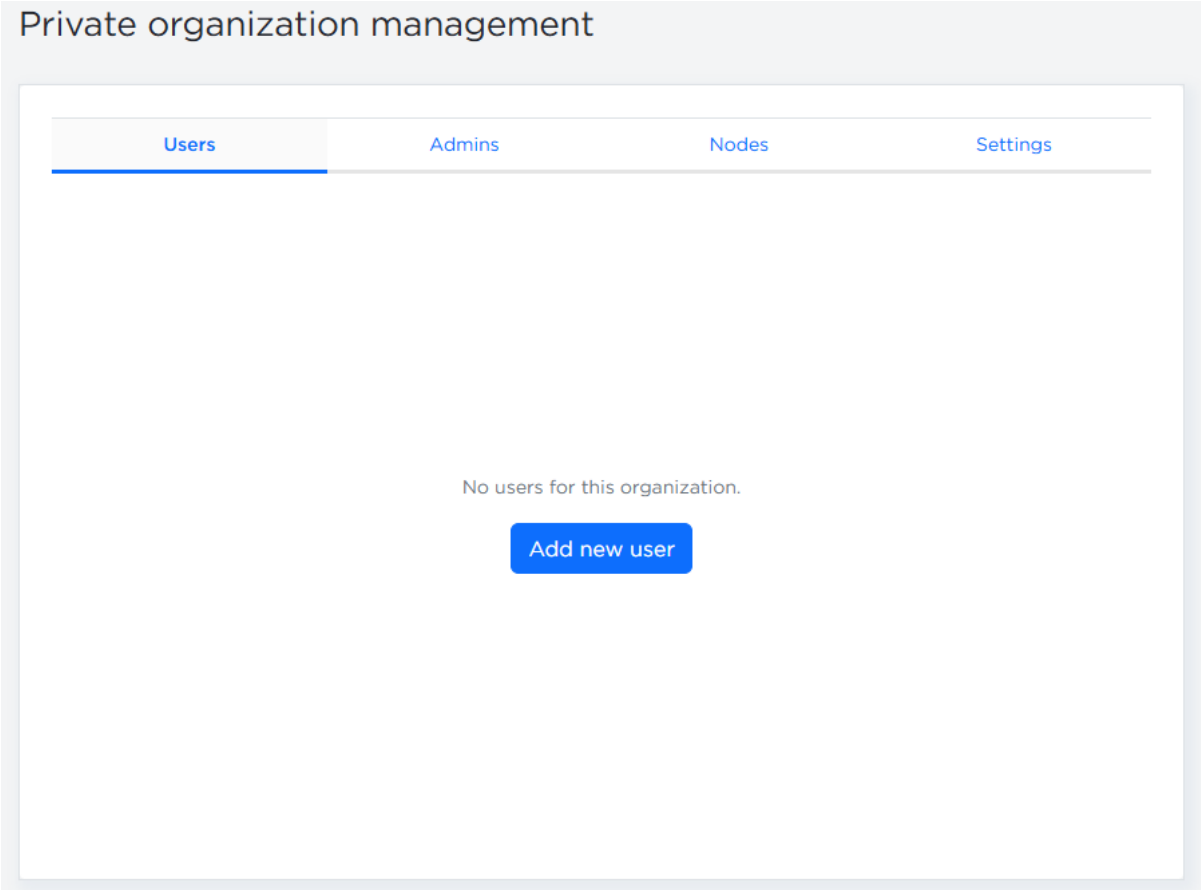
Добавление пользователей в организации

Подобно пользователям в Zabbix, в Zabbix Cloud можно создавать пользователей и назначать им роли, определяющие их права в организации и её нодам.

Чтобы добавить пользователей в организацию:

1. Откройте страницу *Все организации*.

2. Нажмите кнопку настроек нужной организации, чтобы открыть меню управления.



3. На вкладке *Users* или *Admins* нажмите *Add new user*.

4. Введите адрес электронной почты пользователя и выберите роль: *Administrator* или *User*.

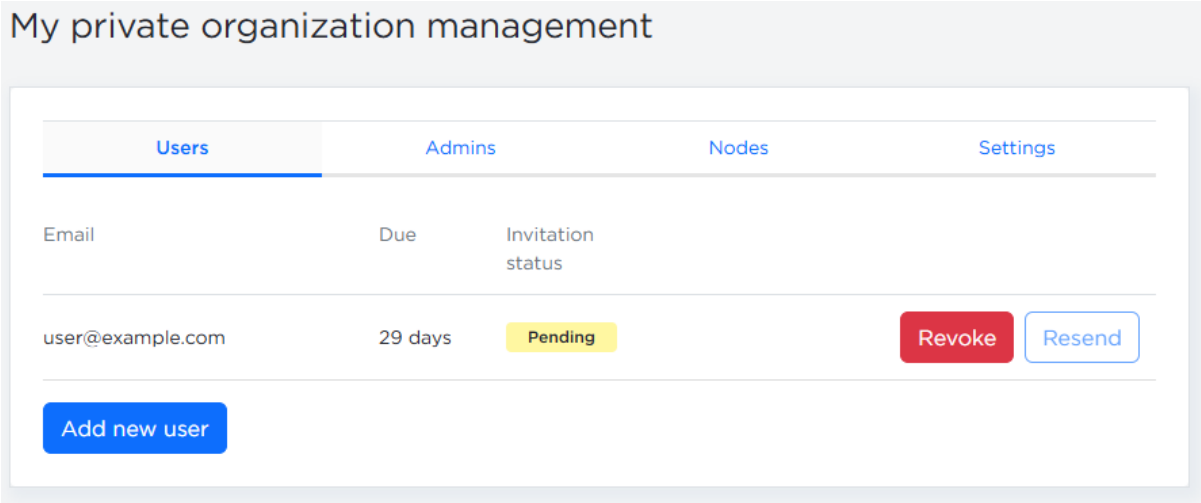
Обратите внимание: выбор роли *User* добавляет пользователя в организацию, но не предоставляет доступ ни к каким нодам.

Каждая роль имеет свой уровень доступа:

- **User** - может работать только с назначенными ему нодами и выполнять конфигурационные задачи, не влияющие на оплату, например:
  - Управление фильтрами доступа
  - Настройка шифрования
  - Задание сроков хранения истории и трендов
  - Планирование технического обслуживания
- **Administrator** - имеет все права пользователя, а также может выполнять действия, влияющие на оплату, например:
  - Создавать и восстанавливать резервные копии
  - Изменять уровень ресурсов ноды (вычислительный уровень и размер диска)
  - Отменять подписки
- **Owner** - полный контроль над организацией. Помимо всех прав администратора, может:
  - Создавать и удалять ноды
  - Добавлять или удалять пользователей и администраторов
  - Переименовывать или удалять организацию
  - Управлять платёжными данными

> **Attention:**
> Платёжный метод владельца используется для всех услуг. Если требуется подтверждение операции банком, именно владелец должен её авторизовать, даже если действие инициировал администратор.

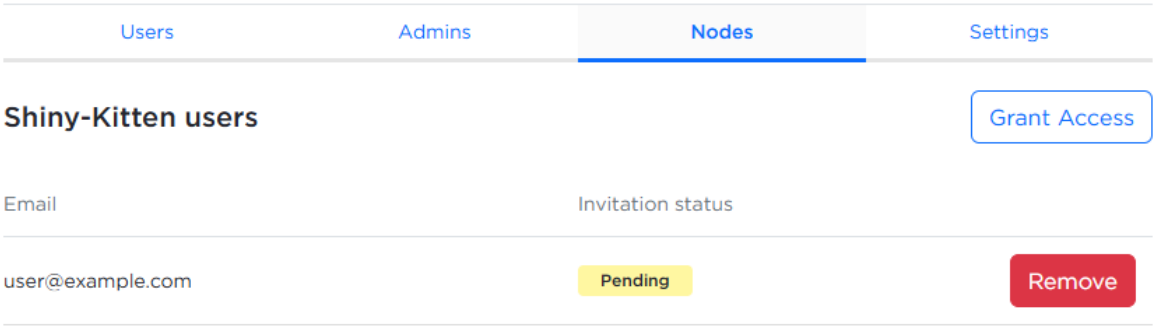
Приглашённый пользователь будет добавлен со статусом *Pending*. Приглашение можно отозвать или отправить повторно (повторная отправка возможна не чаще одного раза в 10 минут — кнопка будет неактивна при превышении лимита).



Добавление пользователей к нодам

Чтобы добавить пользователя к ноде:

1. Перейдите на вкладку *Ноды*.
2. Нажмите *Предоставить доступ*.
3. Выберите пользователя, добавленного в организацию из раскрывающегося списка.
4. Нажмите *Предоставить доступ*.



Кроме того, вы можете предоставить доступ в окне *Конфигурация ноды*:

1. Нажмите кнопку настроек  .
2. Выберите *Добавить пользователя*.
3. Выберите пользователя и нажмите *Предоставить доступ*.

Приглашенные пользователи должны принять приглашение, создав учетную запись Zabbix Cloud, используя тот же адрес электронной почты, на который было отправлено приглашение. Если у пользователя уже есть учетная запись Zabbix Cloud с этим адресом электронной почты, он должен войти в систему и подтвердить приглашение. Приглашения привязаны к определенному адресу электронной почты и не могут быть приняты с другой учетной записи.

Ключевые отличия Zabbix Cloud

Введение

Zabbix Cloud — это облачная платформа, позволяющая пользователям запускать автономные экземпляры Zabbix в облаке (облачные узлы).

По сравнению с обычным локальным развертыванием Zabbix:

- Не требуется установка или обслуживание.
- Облачные узлы полностью управляются Zabbix.
- Облачные узлы можно запустить за несколько минут.

Эта услуга предоставляется по модели оплаты по мере использования (доступна **бесплатная пробная версия**).

Сравнительная таблица Хотя большая часть функциональности локальных установок Zabbix и облачных узлов сети совпадает, между ними есть и различия. Ниже приведена сводная сравнительная таблица. В последующих разделах подробно описаны отдельные элементы данных.

Функция / аспект	Zabbix Cloud	Zabbix локально
Цикл выпуска	Соответствует выпускам LTS	Вы выбираете версию Zabbix (стабильные, бета-версии, пользовательские сборки)
Простота использования	Готовая к использованию платформа, полностью управляемая Zabbix	Требуются установка, настройка и обслуживание
Установка исправлений уязвимостей	Автоматическая установка исправлений для ОС и Zabbix	Пользователь устанавливает исправления вручную
Сертификат HTTPS	Действительный сертификат из коробки (без Let's Encrypt)	Самоподписанный сертификат по умолчанию (можно настроить Let's Encrypt или другой центр сертификации)
Производительность БД (INSERT/UPDATE/SELECT)	Автоматическое секционирование для максимальной скорости; длительные операции DELETE не выполняются	Ручное секционирование и очистка (возможны длительные транзакции DELETE)
Настройка производительности	Предварительно настроена командой Zabbix Cloud (без журналов изменений)	Полная видимость и контроль параметров настройки
Управление брандмауэром	Управляется командой Zabbix Cloud; порты веб-интерфейса и trapper не открыты публично	Полный контроль правил брандмауэра и доступности портов
Исходящий трафик через порт 25	По умолчанию заблокирован для предотвращения спама и злоупотреблений; вместо него используйте порт SMTP 587 или 465	Не ограничивается Zabbix; зависит от вашей сети и поставщика услуг хостинга
Изоляция и защита БД	Высокая степень изоляции, прямой доступ к БД/SSH отсутствует	Зависит от вашей сети и безопасности узла сети
Высокая доступность (HA)	Работает на облачной платформе с высокой доступностью, прокси Zabbix можно использовать для повышения отказоустойчивости сбора данных	HA управляется пользователем
SNMP-трапы	Только через выделенный прокси (приемник трапов без HA/балансировки нагрузки отсутствует)	Напрямую или через прокси; возможны HA и балансировка нагрузки
Опрос SNMP	Затруднен без прокси (для каждого устройства требуется NAT/пользовательский порт)	Встроенный опрос SNMP; прокси необязателен
Настройки хранения данных	История/тренды/аудит настраиваются только через веб-интерфейс (API и файл конфигурации не поддерживаются)	Настраиваются в файле конфигурации сервера Zabbix или через API
Пользовательские скрипты	`AlertScriptsPath`, `ExternalScripts`, модули веб-интерфейса и модули сообщества не поддерживаются	Полностью поддерживаются (пути к скриптам, модули, интеграции)
Мониторинг ODBC	Только PostgreSQL (драйвер `{postgresql}`); плагин MariaDB присутствует как заполнитель	ODBC для PostgreSQL, MySQL, Oracle и других СУБД - полностью настраивается
Ограничение вызовов ODBC	Невозможно (доступен только `StartODBCPollers=1`); ресурсоемкие синхронные запросы блокируют другие операции	Количество опросчиков и расписание полностью настраиваются
Загрузка сертификата SAML	Не поддерживается	Поддерживается через веб-интерфейс или API
Запланированные отчеты	Необходимо создать собственный тип носителя Email (тип носителя «скрипт» не поддерживается)	Из коробки поддерживаются типы носителей «скрипт» и Email

Функция / аспект	Zabbix Cloud	Zabbix локально
Интерфейс узла сети для активных проверок	IP-адрес, предоставленный платформой (создается автоматически)	Интерфейсы управляются вами; IP-адреса находятся под вашим контролем

Различия в функциях SNMP-трапы

SNMP-трапы поддерживаются только через выделенный прокси Zabbix. Если требуется мониторинг SNMP-трапов, автоматическую балансировку нагрузки или высокую доступность для прокси использовать нельзя, поскольку SNMP-трапы должны отправляться на один IP-адрес.

SNMP-опрос

Без прокси SNMP-опрос требует, чтобы каждое устройство было доступно через NAT и использовало настраиваемые порты.

Настройки хранения

Периоды хранения истории, трендов и журнала аудита можно настраивать только через веб-интерфейс в облаке. Эти настройки нельзя задать через `zabbix_server.conf` или API.

Ручные переопределения истории для отдельных элементов данных не поддерживаются (разбиение на разделы управляется глобально). URL-адреса облачных узлов не принимают те же параметры запроса, что и установки on-premise.

Пользовательские скрипты

Следующие типы пользовательских скриптов не поддерживаются в Zabbix Cloud:

- Скрипты оповещений (`AlertScriptsPath`)
- Внешние скрипты (`ExternalScripts`)
- Скрипты веб-интерфейса на Zabbix Server

Модули веб-интерфейса, разработанные сообществом, установить нельзя.

Мониторинг ODBC

Zabbix Cloud поддерживает мониторинг ODBC только для PostgreSQL. Используйте официальный шаблон ODBC Zabbix и задайте строку подключения в шаблоне следующим образом:

```
Driver={postgresql}
```

Также установлен плагин MariaDB, но в настоящее время он не функционирует. Если он используется, задайте:

```
Driver={mysql}
```

Attention:
В Zabbix Cloud нет известной рабочей конфигурации для Oracle MySQL 8.0. Хотя простые запросы (например, `SELECT 1`) могут выполняться успешно, более сложные запросы возвращают `SQL_ERROR`.

 Ограничить вызовы ODBC невозможно (`StartODBCPollers=1` только). Интенсивная синхронная отчетность может повлиять на производительность, и одновременно может выполняться только один `SELECT`.

Доступ к инфраструктуре

Zabbix Cloud не предоставляет SSH-доступ к базовым узлам и не разрешает прямые подключения к базе данных (например, к порту 3306). Все действия по настройке, мониторингу и устранению неполадок должны выполняться через облачный веб-интерфейс или API, что обеспечивает изоляцию и защиту уровней ОС и базы данных.

Исходящий трафик через порт 25

Для предотвращения спама и злоупотреблений исходящие подключения с узлов сети Zabbix Cloud к TCP-порту 25 по умолчанию заблокированы. Это влияет на:

- Типы носителей Email, отправляющие сообщения через SMTP-сервер на порту 25. Поскольку 25 является значением по умолчанию для параметра *Порт SMTP-сервера* нового типа носителя Email, измените его на 587 (при этом для параметра *Безопасность подключения* установите значение STARTTLS) или на 465 (SSL/TLS) в зависимости от поддерживаемых вашим SMTP-сервером возможностей. Поставщик электронной почты *Office365 relay* использовать нельзя, поскольку он подключается к конечной точке ретранслятора Microsoft через порт 25.
- Запланированные отчеты, для которых требуется тип носителя Email.

- Проверки, которые сам узел сети выполняет для порта 25, например простые проверки `net.tcp.service[smtp]` или проверки SMTP в правилах обнаружения сети. При таких проверках служба SMTP отображается как недоступная, даже если она работает; вместо этого выполняйте такие проверки через прокси Zabbix или агент Zabbix в собственной сети.

Стандартный тип носителя *Cloud Email* не затрагивается.

Интерфейс активной проверки

Активные проверки в облаке автоматически создают интерфейс узла сети с IP-адресом, не связанным с вашей локальной сетью. По умолчанию этот IP-адрес может быть внешним. Чтобы сохранить согласованность, вы можете вручную удалить или изменить интерфейс узла сети после его создания.

Загрузка сертификата SAML

SAML authentication не поддерживается, поскольку немногие поставщики принимают незашифрованные или неподписанные сообщения SAML.

Запланированные отчёты

Медиа-тип *Cloud Email* по умолчанию использует транспорт скрипта и не может применяться для запланированных отчётов. Чтобы отправлять отчёты по электронной почте:

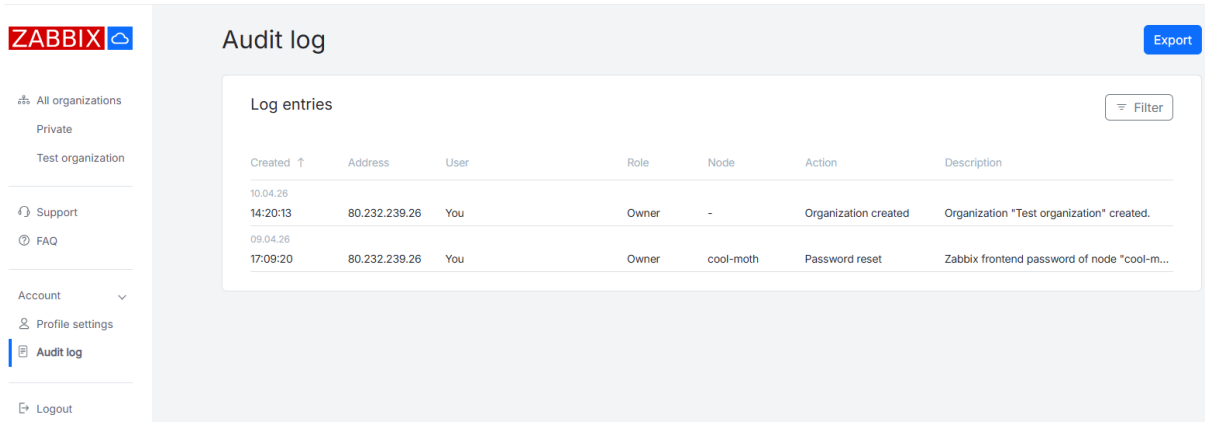
1. Создайте новый медиа-тип Email.
2. Назначьте медиа электронной почты своему пользователю в разделе *Настройки пользователя* > *Мультимедиа*.
3. Выберите этот медиа-тип при настройке запланированных отчётов.

Attention:
Не используйте порт 25 в качестве *порта SMTP-сервера* для этого медиа-типа; см. раздел Исходящий трафик через порт 25.

Журнал аудита

Обзор В Zabbix Cloud журнал аудита записывает действия пользователей и системы.

После входа в систему в учетную запись Zabbix Cloud перейдите в *Account* > *Audit log*, чтобы просмотреть записи журнала аудита. Доступные пользователю записи журнала аудита зависят от роли пользователя в организации.



Просмотр записей журнала аудита Записи журнала аудита отображаются в таблице.

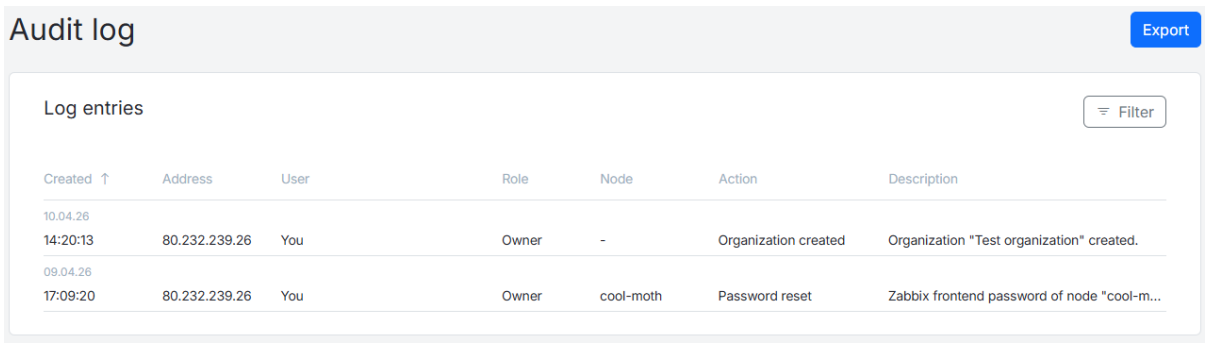


Таблица содержит следующую информацию:

Column	Description
Created	Дата и время создания записи журнала. Отображаемое время соответствует часовому поясу пользователя, выбранному в выпадающем списке *User timezone* в *Account > Profile settings*.
User	Имя пользователя организации, который выполнил действие.
Role	Роль пользователя, который выполнил действие.
Node	Имя узла, связанного с действием.
Action	Имя выполненного действия.
Description	Описание выполненного действия.

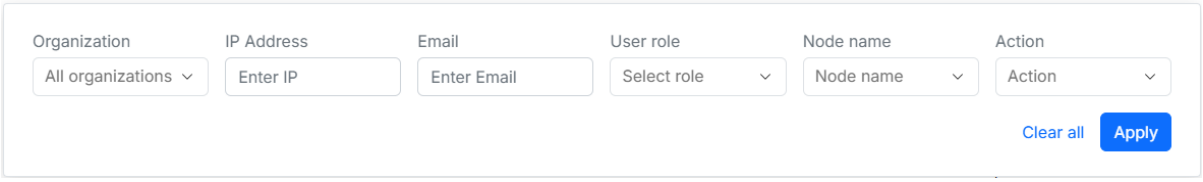
По названиям столбцов можно щелкнуть, чтобы отсортировать записи по возрастанию или убыванию.

Отфильтрованные записи журнала аудита можно экспортировать в формате `.csv` с помощью кнопки *Export*.

Использование фильтров

Нажмите кнопку *Filter*, чтобы открыть фильтр.

Чтобы отобразить отфильтрированные результаты, после выбора одного или нескольких параметров фильтрации нажмите *Apply*. Чтобы очистить фильтр, нажмите *Clear all*.



Записи можно фильтровать по одному или нескольким параметрам. Доступны следующие параметры фильтрации:

Параметр	Описание
Organization	Выберите все организации или конкретную организацию.
IP Address	Укажите IP-адрес.
Email	Укажите адрес электронной почты, связанный с пользователем.
User role	Выберите все роли или конкретную роль: Owner, Admin или User.
Node name	Выберите все узлы сети или конкретный узел сети.
Action	Выберите конкретный тип действия.

В журнале аудита отслеживаются следующие действия:

- *Создание узла сети*
- *Удаление узла сети*
- *Обновление узла сети*
- *Обновление версии узла сети*
- *Сброс пароля*
- *Обновление параметров обслуживания*
- *Обновление параметров истории*
- *Продление даты окончания срока действия узла сети*
- *Обновление конфигурации прокси*
- *Создание резервной копии*
- *Удаление резервной копии*
- *Восстановление резервной копии*
- *Обновление резервной копии*
- *Добавление фильтра доступа*
- *Обновление фильтра доступа*
- *Удаление фильтра доступа*
- *Включение шифрования*
- *Отключение шифрования*
- *Отмена подписки*
- *Продление подписки*
- *Сбой платежа по подписке*
- *Платеж требует подтверждения*

- *Создание организации*
- *Удаление организации*
- *Приглашение администратора*
- *Приглашение пользователя*
- *Присоединение администратора*
- *Присоединение пользователя*
- *Удаление администратора из организации*
- *Удаление пользователя из организации*
- *Добавление узла сети в организацию*
- *Удаление узла сети из организации*
- *Предоставление доступа к узлу сети*
- *Отзыв доступа к узлу сети*
- *Обновление настроек пользователя*
- *Создание платежа*
- *Завершение платежа*
- *Сбой платежа*
- *Начисление НДС*
- *Сбой проверки НДС*
- *Обновление платежной информации*
- *Обновление способа оплаты*
- *Удаление способа оплаты*
- *Создание узла сети с предоплатой*
- *Обновление узла сети с предоплатой*
- *Начало инцидента*
- *Завершение инцидента*

Настройка облачного прокси Zabbix

Обзор На этой странице описывается настройка прокси Zabbix, развернутого в Zabbix Cloud, для сбора данных мониторинга с узлов сети в удаленной среде.

Процедура предназначена главным образом для сервера Zabbix и прокси, созданных в Zabbix Cloud.

В процедуре также описывается конфигурация, необходимая, если сервер Zabbix размещен не в Zabbix Cloud.

Предварительные требования Установлен, настроен и запущен сервер Zabbix либо на Zabbix Cloud работает узел сервера Zabbix.

Создание узла прокси Создайте узел прокси Zabbix Cloud.

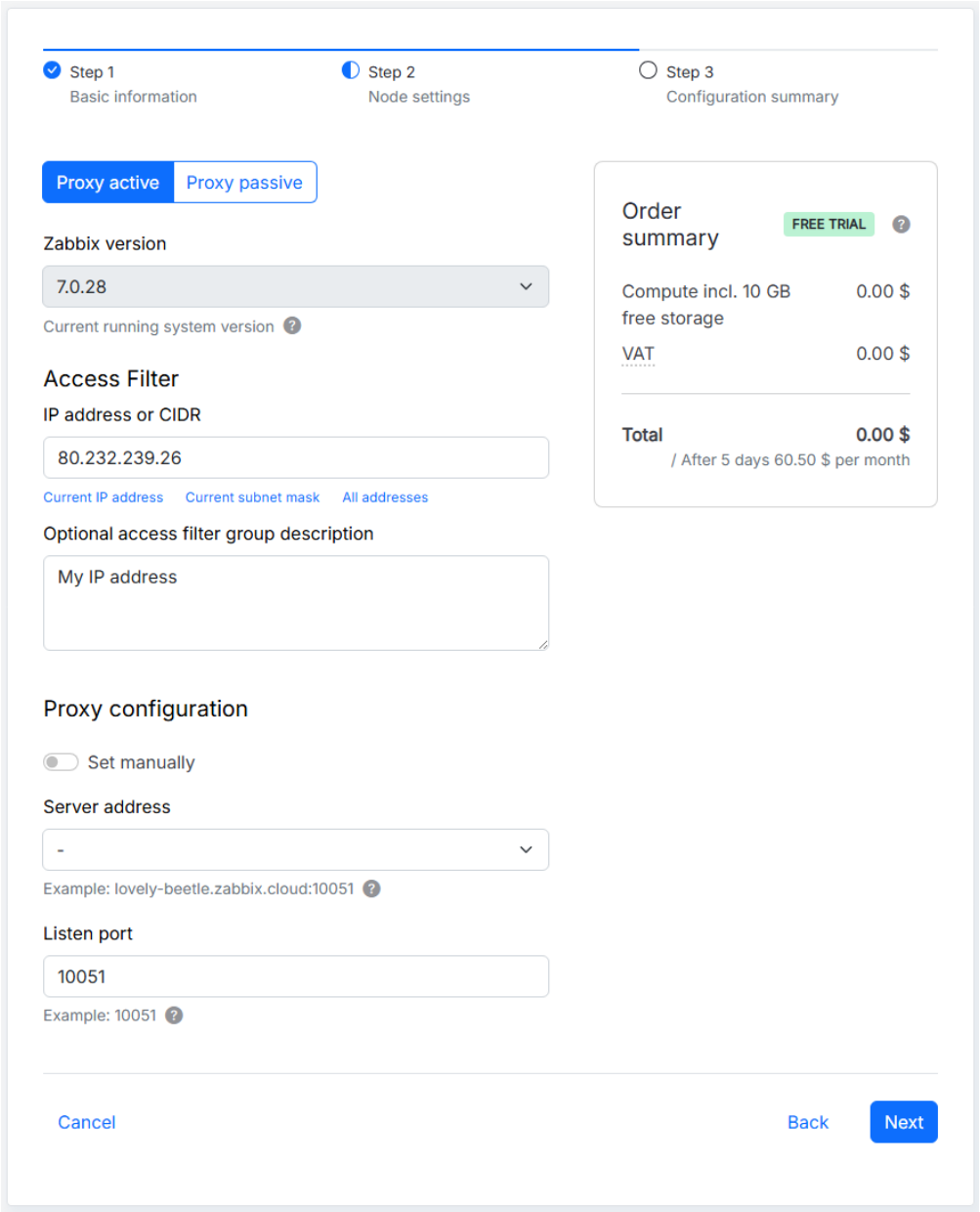
При создании узла:

1. Выберите *Proxy*.

2. Выберите *Proxy active* или *Proxy passive* в зависимости от того, как прокси должен взаимодействовать с сервером Zabbix.

3. Настройте IP-адрес узла или фильтры доступа CIDR (это можно сделать после создания узла).

4. Укажите сервер Zabbix, к которому будет подключаться прокси.



5. Завершите процесс создания узла.

Дополнительную информацию о создании узла прокси см. в разделе Развертывание Zabbix в облаке.

Настройка фильтров доступа Zabbix Cloud использует фильтры доступа, чтобы контролировать, какие IP-адреса или подсети могут подключаться к облачному узлу.

Во время инициализации узла автоматически создается фильтр доступа для IP-адреса, используемого для доступа к узлу. Он не обязательно включает адреса отслеживаемых узлов сети или других компонентов Zabbix. Добавьте необходимые адреса вручную.

Дополнительную информацию см. в разделе Конфигурация узла — Фильтры доступа.

Узел прокси

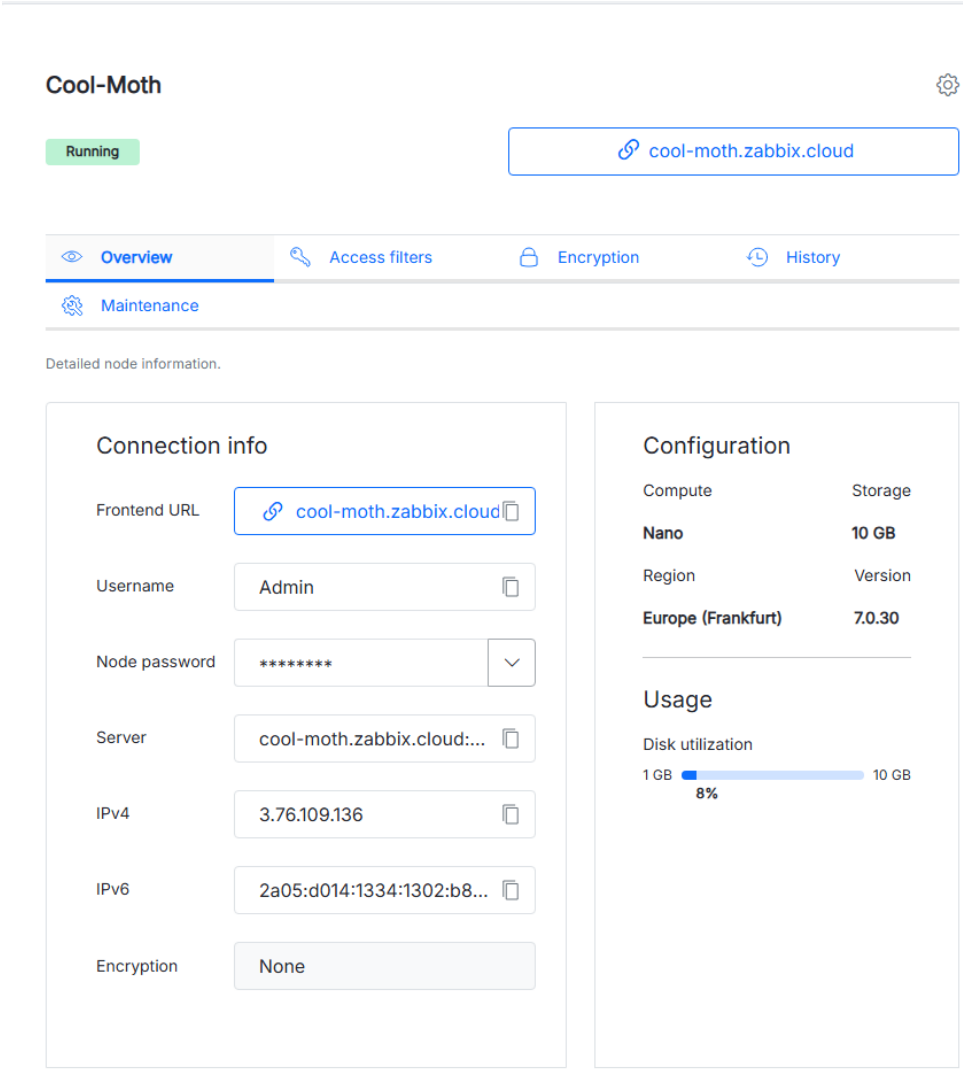
Добавьте IP-адреса или подсети, с которых ожидаются подключения к прокси.

Если отслеживаемые узлы сети используют активные проверки агента, агенты инициируют подключения к прокси, поэтому IP-адрес или подсеть отслеживаемых узлов сети должны быть разрешены фильтрами доступа узла прокси.

Если в качестве режима прокси был установлен *Proxy Passive*, сервер Zabbix подключается к прокси, поэтому IP-адрес или подсеть сервера Zabbix должны быть разрешены фильтрами доступа узла прокси.

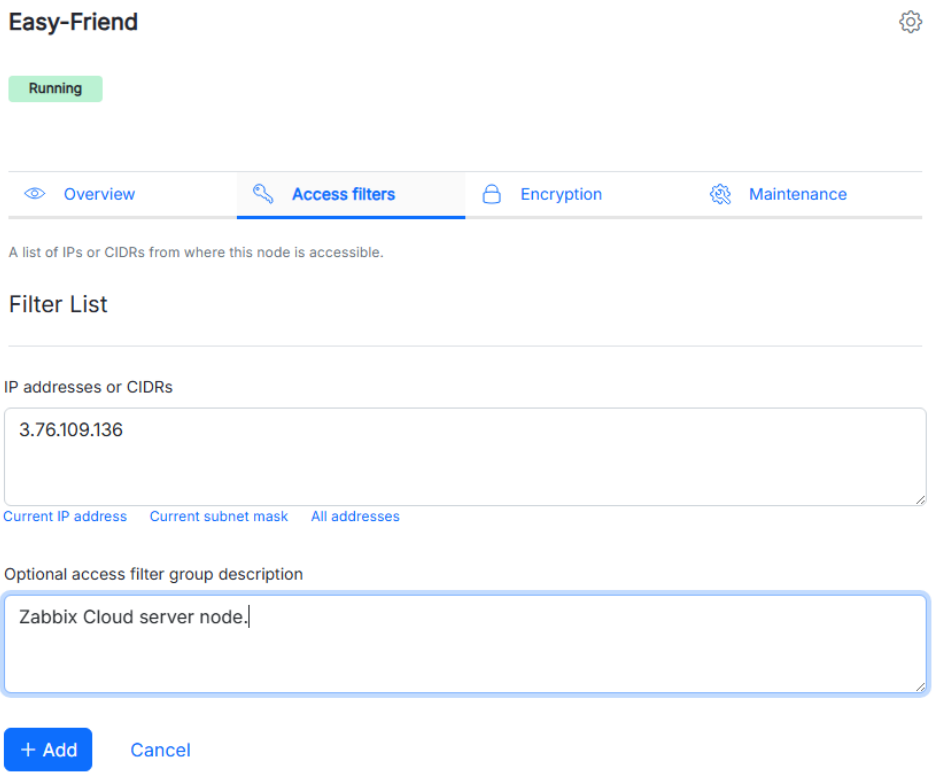
Если выбран режим *Proxy passive* и используется узел сервера:

1. Перейдите на вкладку *Overview* узла сервера и скопируйте IPv4- или IPv6-адрес узла.

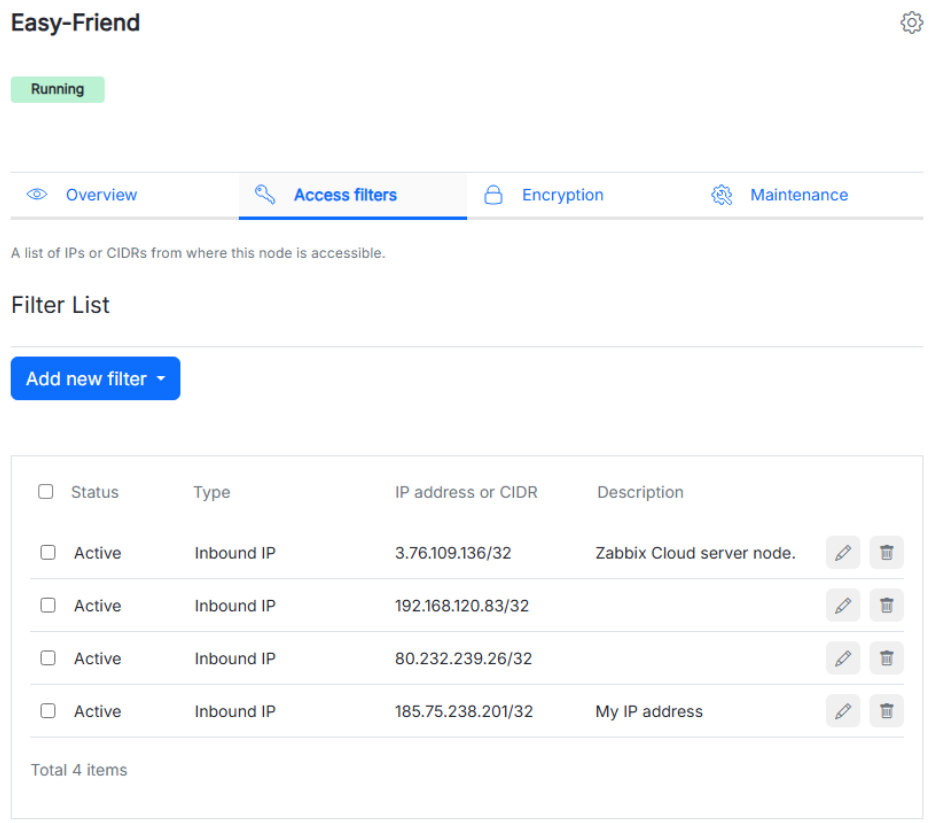


2. Перейдите на вкладку *Access filters* узла прокси и нажмите *Add new filter* > *Add*.

3. Вставьте скопированный адрес в поле *IP addresses or CIDRs* и добавьте описание в поле *Optional access filter group description*.



4. Нажмите *Add*.



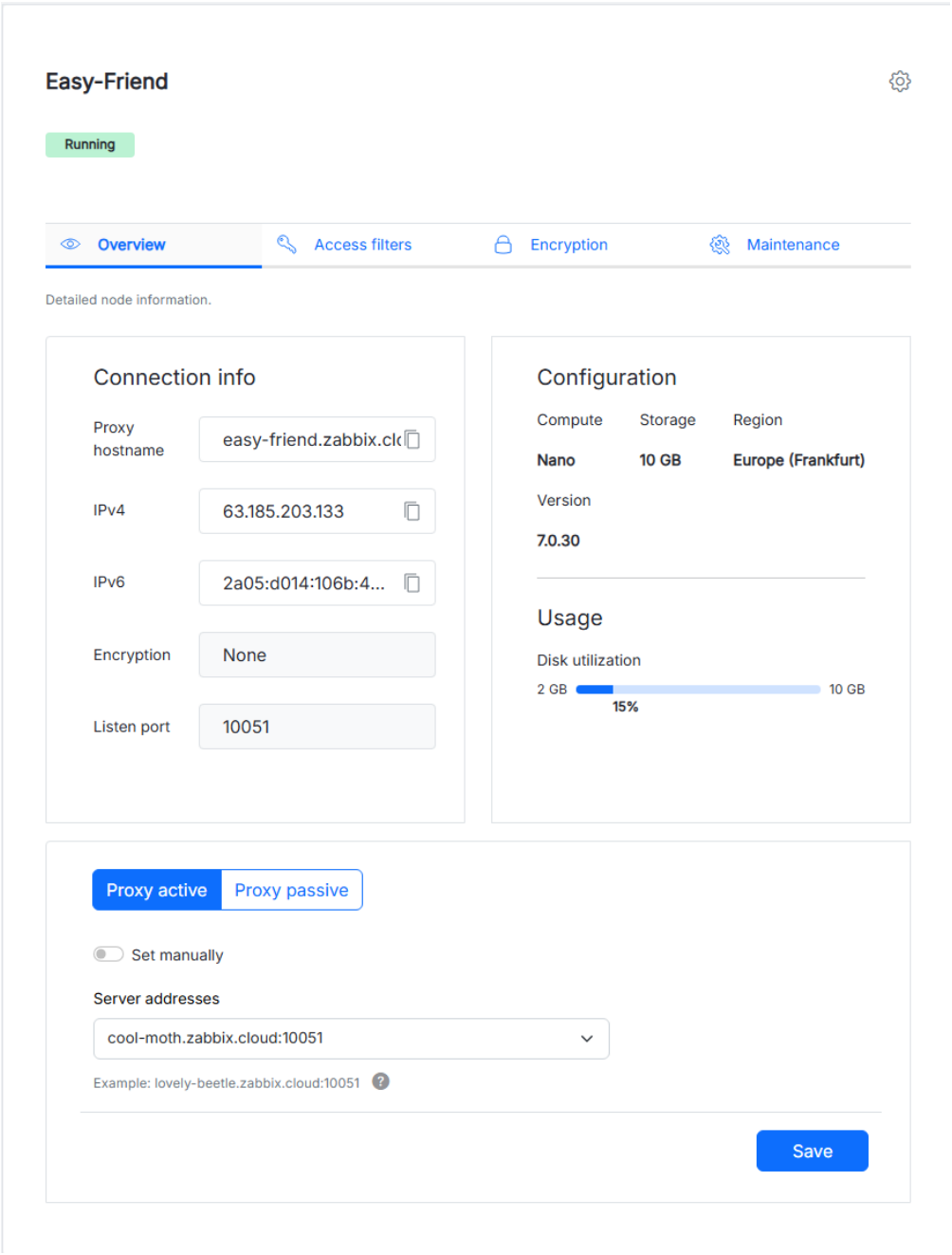
Выполните шаги 2-4 для IP-адреса агента Zabbix, использующего активные проверки.

Узел сервера

Если сервер Zabbix размещен в Zabbix Cloud, также настройте фильтры доступа на узле сервера.

Для прокси в активном режиме добавьте IP-адрес или CIDR прокси-узла в фильтры доступа узла сервера, поскольку прокси инициирует подключение к серверу.

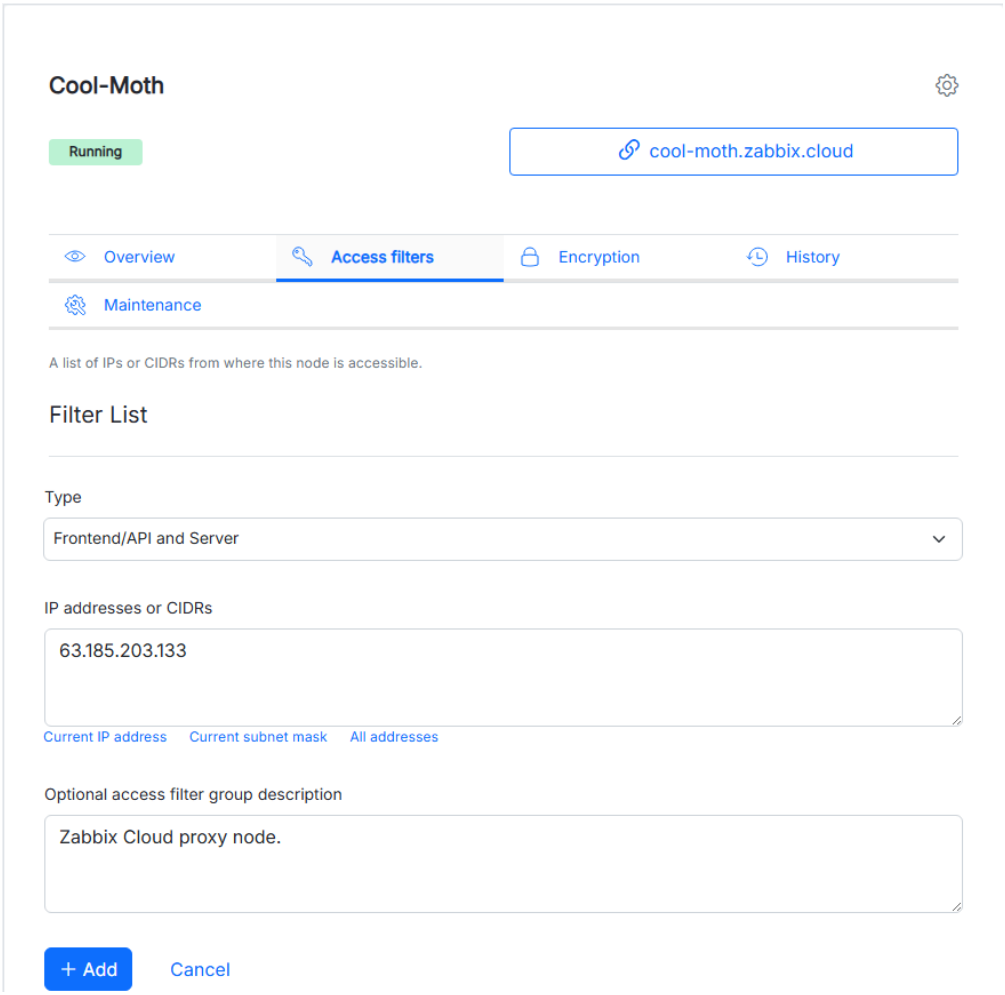
1. Перейдите на вкладку *Обзор* прокси-узла и скопируйте IPv4- или IPv6-адрес узла.



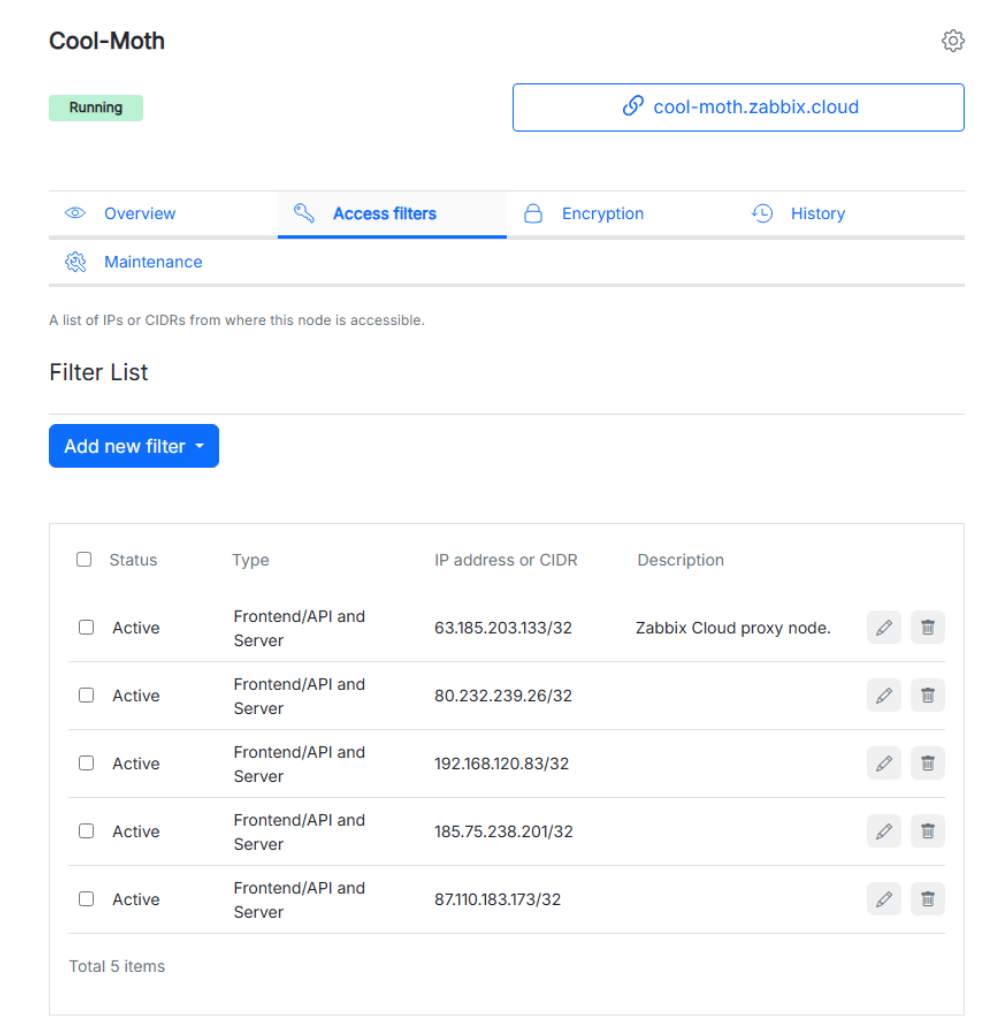
2. Перейдите на вкладку *Фильтры доступа* узла сервера и нажмите *Добавить новый фильтр* > *Добавить*.

3. Вставьте скопированный адрес в поле *IP-адреса или CIDR*, а в поле *Необязательное описание группы фильтров доступа* добавьте описание.

4. Выберите подходящий тип фильтра.



5. Нажмите *Добавить*.



Настройка прокси Прокси необходимо настроить в веб-интерфейсе Zabbix для взаимодействия с сервером Zabbix.

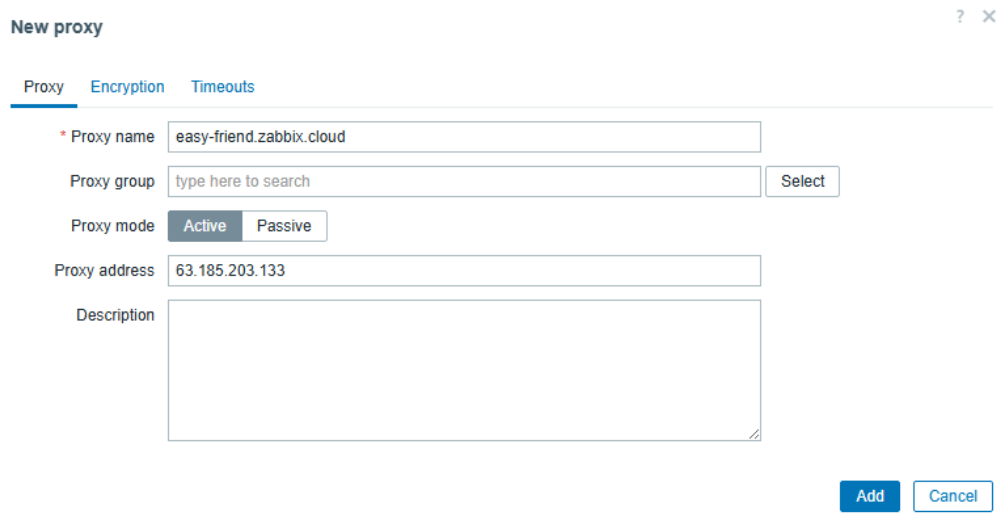
Откройте раздел *Администрирование* > *Прокси* в веб-интерфейсе Zabbix, нажмите *Создать прокси* и настройте прокси необходимым образом:

1. Укажите имя прокси в поле *Имя прокси*. Оно должно совпадать с именем узла сети прокси, настроенным для прокси. Введите его вручную или скопируйте на вкладке *Обзор* прокси.

2. Выберите режим прокси, соответствующий режиму, выбранному при создании прокси.

3. Для активного прокси укажите IP-адрес или DNS-имя прокси.

4. Нажмите *Добавить*.



Дополнительную информацию см. в разделе Распределенный мониторинг с использованием прокси.

Настройка отслеживаемых узлов сети Добавьте узлы сети, которые должны отслеживаться через прокси.

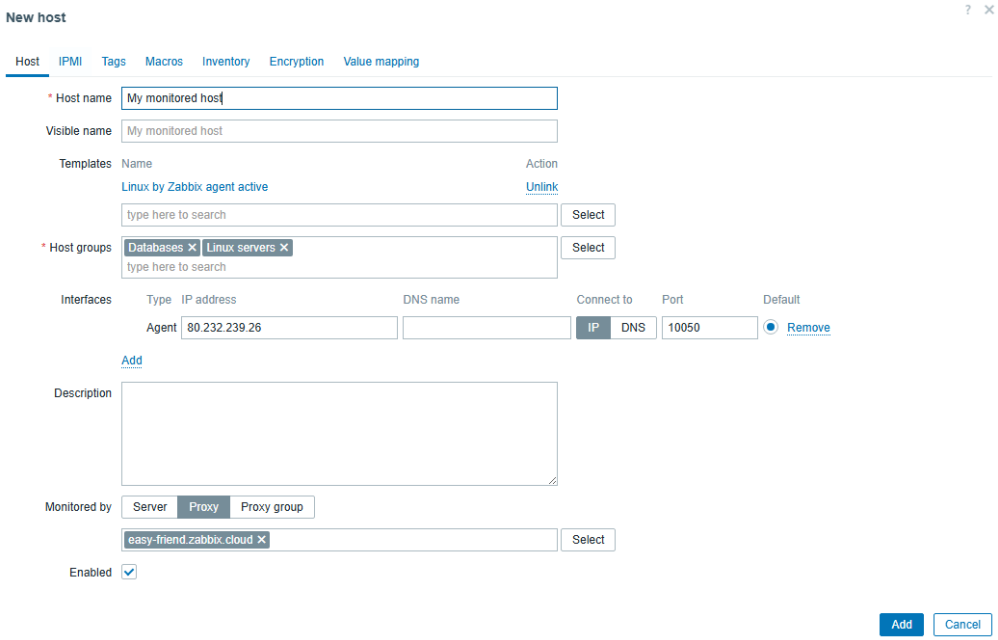
1. Перейдите в раздел *Сбор данных* > *Узлы сети*.

2. Создайте узел сети или откройте существующий.

3. В поле *Отслеживается через* выберите *Прокси*: введите имя прокси для поиска и выберите его либо нажмите кнопку *Выбрать*, чтобы открыть список доступных прокси.

4. Настройте интерфейс узла сети и другие параметры.

5. Нажмите *Добавить* или *Обновить*.



Узел сети, настроенный для отслеживания через прокси, отправляет или получает данные мониторинга через этот прокси в зависимости от типа элемента данных и режима агента.

Настройка агента Zabbix Если на узле сети, за которым ведется наблюдение, используется агент Zabbix или агент Zabbix 2, настройте агент для взаимодействия с соответствующим компонентом Zabbix.

Конфигурация зависит от того, использует ли агент *пассивные* или *активные* проверки.

Пассивные проверки агента

При пассивных проверках прокси Zabbix инициирует подключения к агенту и запрашивает данные мониторинга.

Откройте файл конфигурации агента Zabbix и измените параметр `Server`, чтобы разрешить подключения с IP-адреса прокси Zabbix Cloud.

Например:

```
Server=<proxy_IP>
```

Убедитесь, что узел сети с мониторингом разрешает входящие TCP-подключения от прокси через порт агента, который по умолчанию имеет значение 10050.

Прокси должен иметь возможность подключаться к узлу сети с мониторингом по сети.

Проверки активным агентом

При активных проверках агент инициирует подключения к прокси Zabbix и отправляет собранные данные мониторинга.

Откройте файл конфигурации агента Zabbix и обновите параметр `ServerActive`, указав адрес прокси Zabbix Cloud.

Например:

`ServerActive=<proxy_IP>:<proxy_listen_port>`

Замените `<proxy_IP>` на IP-адрес прокси Zabbix Cloud, а `<proxy_listen_port>` — на порт прослушивания, настроенный при создании узла сети прокси. Порт прослушивания по умолчанию — 10051.

Если в параметре `ServerActive` указан прокси, агент получает конфигурацию активных проверок от прокси, а не напрямую от сервера Zabbix.

Убедитесь, что отслеживаемый узел сети может установить исходящее TCP-соединение с прокси через настроенный порт прослушивания.

Дополнительную информацию о пассивных и активных проверках агентом см. в разделе Пассивные и активные проверки агентом.

Внешний сервер Zabbix Прокси Zabbix Cloud также может взаимодействовать с сервером Zabbix, развернутым за пределами Zabbix Cloud.

В этом случае внешний сервер Zabbix должен иметь возможность взаимодействовать с облачным прокси в соответствии с выбранным режимом прокси.

Для прокси в активном режиме прокси инициирует подключение к внешнему серверу Zabbix. Настройте внешний сервер Zabbix на прием подключений от прокси и убедитесь, что прокси может подключиться к серверу по сети.

Для пассивного прокси подключение к прокси инициирует сервер Zabbix. Добавьте IP-адрес или подсеть внешнего сервера Zabbix в фильтры доступа узла сети прокси и убедитесь, что сервер может подключиться к прокси по сети.

Внешний сервер Zabbix не использует фильтры доступа Zabbix Cloud. Вместо этого настройте необходимые правила брандмауэра и сети во внешней среде.

Контролируемые узлы сети могут находиться в той же среде, что и внешний сервер Zabbix, в другой сети или в сети, доступной прокси.

Если агенты используют активные проверки, настройте их параметр `ServerActive`, указав облачный прокси Zabbix. Если агенты используют пассивные проверки, настройте их параметр `Server`, разрешив подключения от прокси.

Проверьте конфигурацию После настройки прокси, агентов, фильтров доступа и сетевого подключения убедитесь, что прокси взаимодействует с сервером Zabbix.

1. Перейдите в раздел *Администрирование* > *Прокси*.

2. Проверьте *Состояние* прокси.

3. Перейдите в раздел *Мониторинг* > *Последние данные* и убедитесь, что доступны последние значения элементов данных.

Если прокси не взаимодействует с сервером, проверьте:

- режим прокси;
- адрес и порт сервера;
- фильтры доступа прокси;
- фильтры доступа сервера, если сервер размещен в Zabbix Cloud;
- конфигурацию агента;
- правила брандмауэра и маршрутизации;
- адрес, настроенный в `Server` или `ServerActive`.

Мобильное приложение

Мобильное приложение

Мобильное приложение Zabbix позволяет контролировать среду Zabbix с мобильного устройства. Просматривайте проблемы, проверяйте их подробности, обновляйте проблемы и получайте push-уведомления, когда требуется ваше внимание.

Получить приложение (скоро)

Настройка Zabbix для приложения

Настройка адаптера моста Zabbix Установите адаптер моста Zabbix, чтобы мобильные устройства могли подключаться к Zabbix.

Добавление устройств Добавьте мобильное устройство в веб-интерфейсе Zabbix, чтобы подключить его к серверу Zabbix.

Настройка push-уведомлений Получайте оповещения на мобильном устройстве при возникновении проблемы.

Узнайте о приложении

Обзор приложения Изучите основы использования мобильного приложения Zabbix.

Безопасный мобильный доступ Защитите приложение, мобильные устройства и адаптер Zabbix bridge или отзовите доступ для утерянных или скомпрометированных устройств.

Настройка Zabbix для приложения

Чтобы использовать мобильное приложение Zabbix (скоро будет доступно), сначала необходимо:

1. Настроить адаптер моста Zabbix.
2. Настроить push-уведомления.
3. Добавить мобильное устройство в веб-интерфейсе Zabbix.

В разделах ниже описан каждый шаг.

После настройки ознакомьтесь с разделом Безопасный мобильный доступ, чтобы обеспечить безопасность приложения и адаптера моста Zabbix.

Настройка адаптера моста Zabbix Чтобы сервер Zabbix мог взаимодействовать с мобильным приложением, необходимо настроить адаптер моста Zabbix.

Адаптер моста можно установить в той же системе, что и сервер Zabbix, или в другой системе с помощью пакетов:

> **Attention:**
> Ниже приведены инструкции для Ubuntu 24.04. Для других операционных систем используйте команды установки репозитория и пакетов (например, `wget`, `dpkg`, `apt`) со страницы Получение Zabbix, соответствующей вашей ОС, и установите пакет `zabbix-bridge-adapter`.

1. Запустите новый сеанс оболочки с правами root:

```
sudo -s
```

2. Установите репозиторий Zabbix:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Установите адаптер моста Zabbix:

```
apt install zabbix-bridge-adapter
```

4. Отредактируйте файл конфигурации сервера Zabbix:

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Перезапустите сервер Zabbix, например (см. другие варианты):

```
systemctl restart zabbix-server
```

6. Скопируйте идентификатор сервера Zabbix в разделе *Отчеты* > *Сведения о системе*; он понадобится на следующем шаге.

7. Отредактируйте файл конфигурации адаптера моста Zabbix:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

> **Attention:**
> Каталог, указанный в `DataDir`, должен иметь права доступа 0700, а все хранящиеся в нем файлы должны иметь права доступа 0600.

8. Ознакомьтесь с лицензионным соглашением конечного пользователя (EULA) адаптера моста Zabbix и примите его:

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Запустите адаптер моста Zabbix:

```
systemctl start zabbix-bridge-adapter
```

Attention:
При запуске адаптер моста Zabbix использует идентификатор сервера, чтобы зарегистрировать ваш сервер Zabbix в централизованно размещенном Zabbix Bridge. Состояние регистрации хранится в файле состояния в `DataDir`. Если переустановить адаптер моста и файл состояния будет утерян, повторная регистрация того же сервера Zabbix может завершиться ошибкой. Чтобы избежать этого, создайте резервную копию файла состояния перед переустановкой адаптера моста и восстановите ее после переустановки.

Настройка push-уведомлений Чтобы получать оповещения на мобильном устройстве, когда в контролируемой среде требуется внимание:

1. Создайте тип носителя push-уведомлений (или включите тип по умолчанию).
2. Создайте действие (или включите действие триггера *Сообщать о проблемах администраторам Zabbix* по умолчанию).
3. Добавьте устройство, чтобы автоматически добавить тип носителя вашему пользователю (или добавьте носитель пользователя вручную).

Добавление устройств Чтобы использовать мобильное приложение Zabbix на своем устройстве, необходимо добавить это устройство в веб-интерфейсе Zabbix. Инструкции см. в следующих разделах:

- *Пользователи* > *Устройства* (в роли *суперадминистратора* можно добавлять собственные мобильные устройства или мобильные устройства других пользователей)
- *Настройки пользователя* > *Устройства* (любой пользователь может добавлять собственные мобильные устройства)

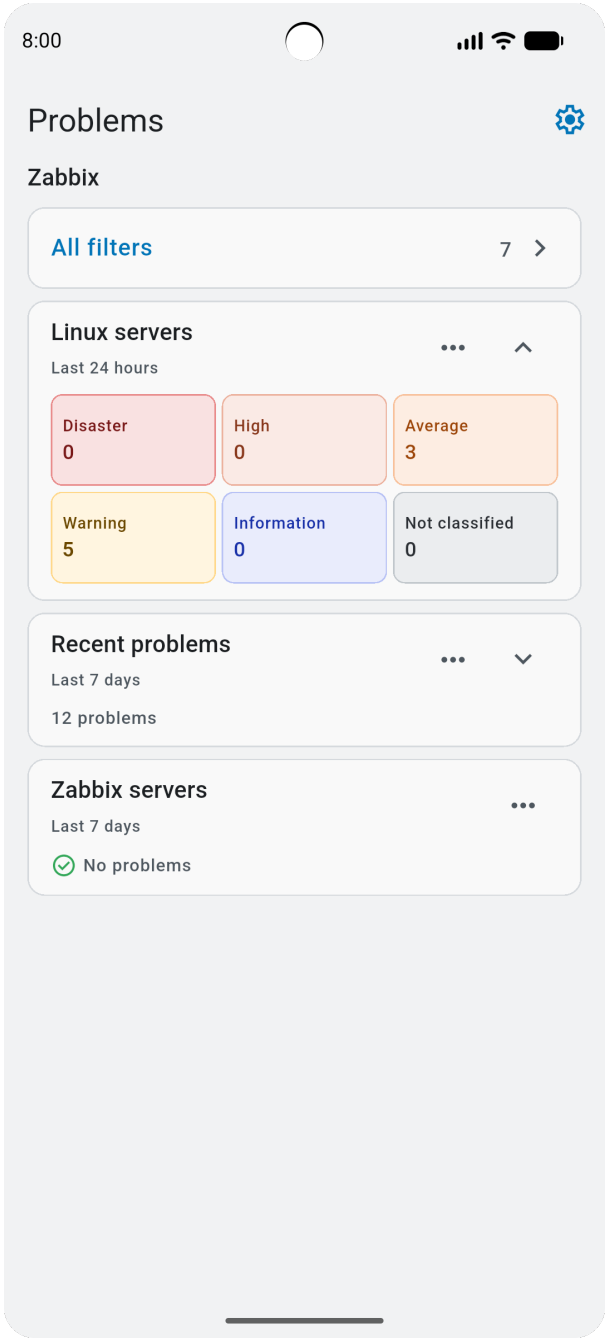
Обзор приложения

Мобильное приложение Zabbix (скоро будет доступно) позволяет контролировать среду Zabbix с мобильного устройства. Просматривайте проблемы, проверяйте их сведения, обновляйте проблемы и получайте push-уведомления, когда требуется ваше внимание.

На этой странице представлен обзор мобильного приложения и описано, как его использовать.

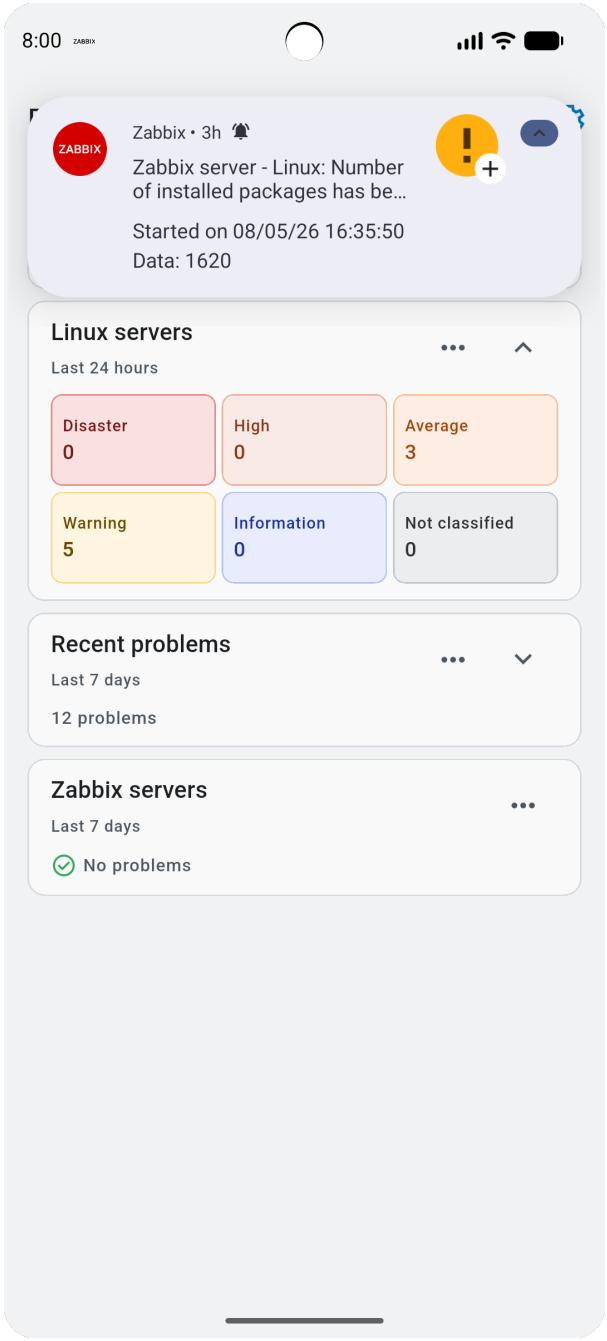
Проблемы На экране *Проблемы* отображается обзор проблем на основе закрепленных фильтров. Он содержит:

- Значок *Настройки* - нажмите его, чтобы просмотреть настройки приложения.
- *Все фильтры* - нажмите, чтобы просмотреть фильтры и управлять ими.
- Имя сервера Zabbix (или *Zabbix*, если оно не было настроено при установке веб-интерфейса Zabbix).
- Закрепленные фильтры - нажмите имя фильтра или уровень серьезности, чтобы открыть экран *Список проблем*. Нажмите меню с тремя точками, чтобы управлять фильтром (редактировать, дублировать, открепить или удалить его), либо нажмите стрелку, чтобы развернуть или свернуть фильтр. Обратите внимание, что закрепленные фильтры упорядочены по алфавиту, а фильтр *Недавние проблемы* нельзя редактировать или удалить.



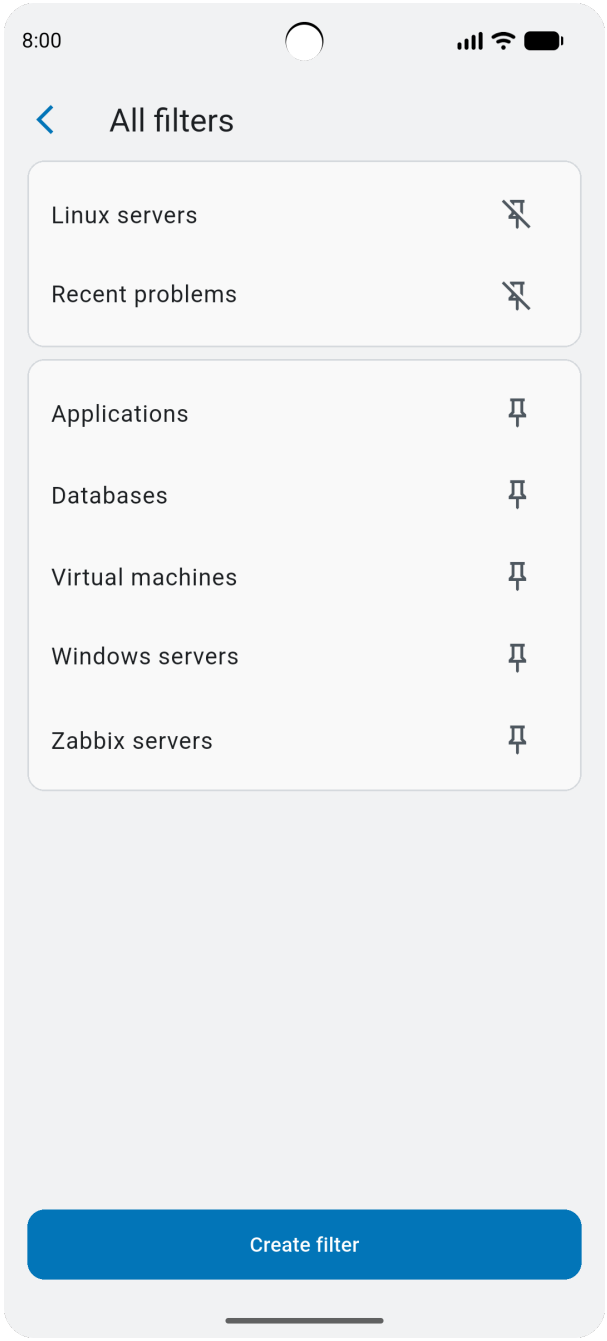
Уведомления Когда сервер Zabbix обнаруживает проблему, обновляет или устраняет ее, он может отправить уведомление на ваше мобильное устройство. Уведомление содержит узел сети с проблемой, ее название и сообщение, сформированное на основе шаблонов сообщений, настроенных для типа носителя push-уведомление.

Нажмите на уведомление, чтобы открыть экран *Сведения о проблеме*.



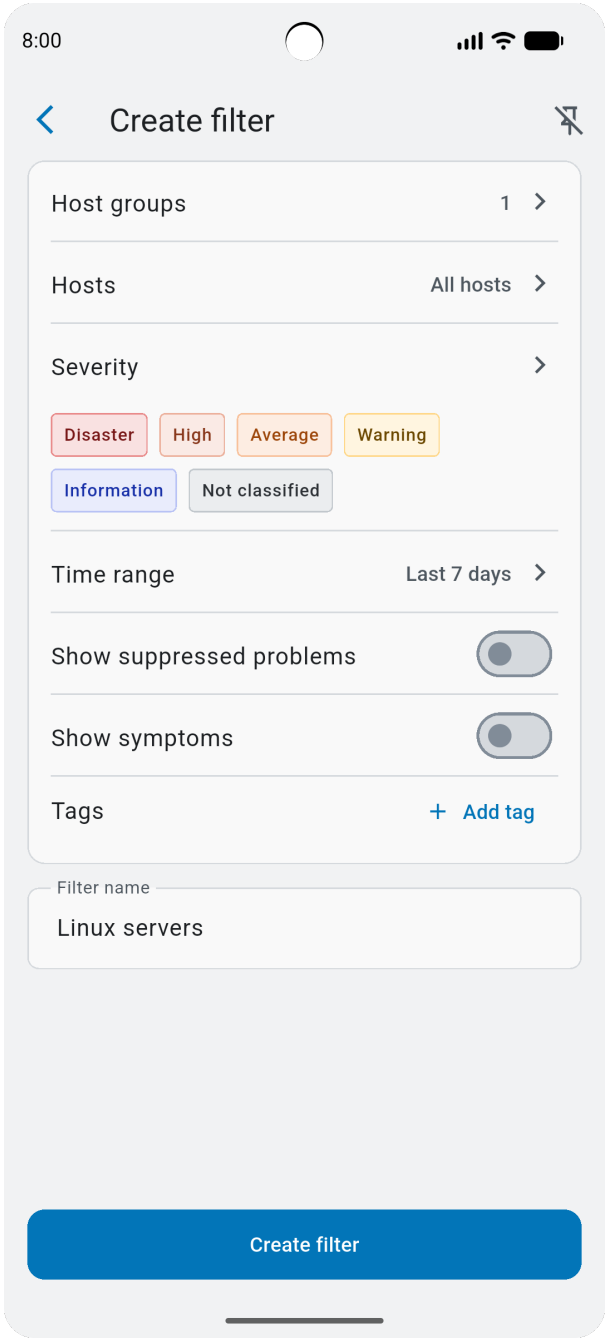
Все фильтры Экран *Все фильтры* позволяет управлять фильтрами. Он содержит:

- Названия фильтров — нажмите, чтобы изменить фильтр.
- Значки закрепления — нажмите, чтобы закрепить фильтр на экране *Проблемы* или открепить его.
- *Создать фильтр* — нажмите, чтобы создать новый фильтр.



Создание фильтра Экран *Создание фильтра* позволяет создать фильтр с параметрами, аналогичными параметрам в разделе *Мониторинг* > *Проблемы* (см. подробности). Также можно задать *Временной диапазон*, в течение которого будут отображаться нерешённые проблемы.

Нажмите *Создать фильтр*, чтобы сохранить фильтр. Новые фильтры закрепляются на экране *Проблемы*; нажмите значок булавки, чтобы изменить это.



Список проблем На экране *Список проблем* отображаются проблемы, соответствующие фильтру (например, *Недавние проблемы*).

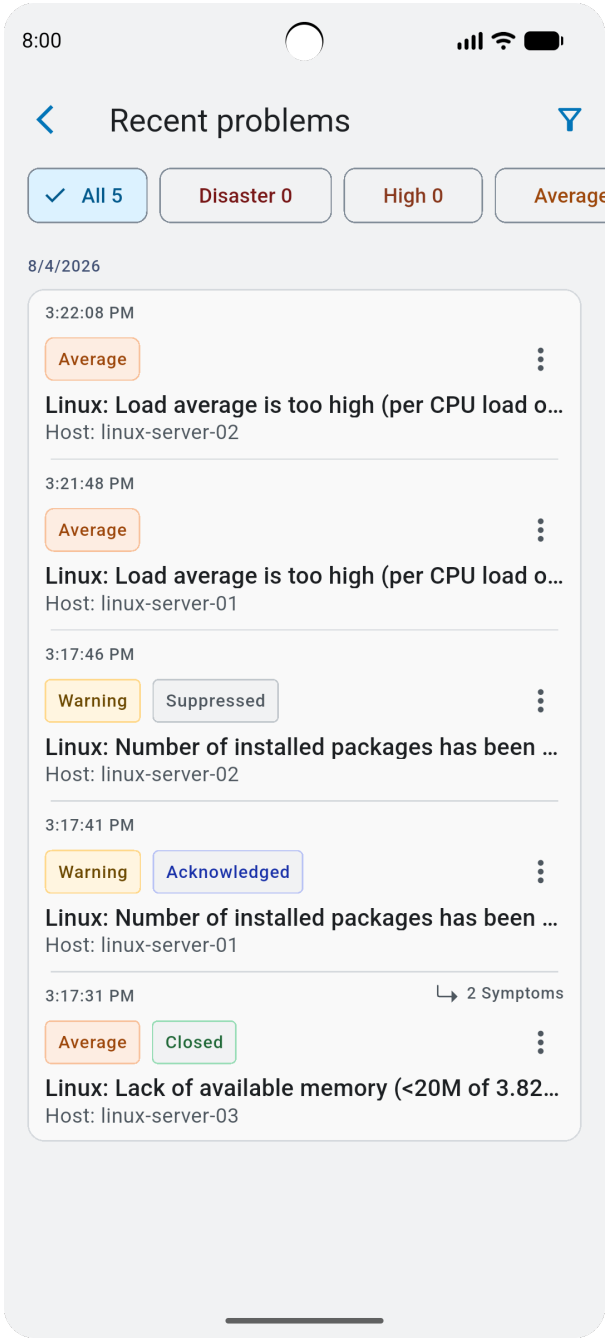
Нажмите значок фильтра или уровень серьезности проблемы, чтобы дополнительно сузить список.

Для каждой проблемы можно увидеть:

- Время и дату начала.
- Уровень серьезности, имя и узел сети.
- Была ли она подавлена (отображается, если включен параметр *Показывать подавленные проблемы*).
- Была ли она подтверждена или недавно закрыта (устранена автоматически или закрыта вручную).
- Есть ли у нее симптомы - нажмите, чтобы просмотреть список проблем-симптомов.

Нажмите на проблему, чтобы открыть экран *Сведения о проблеме*.

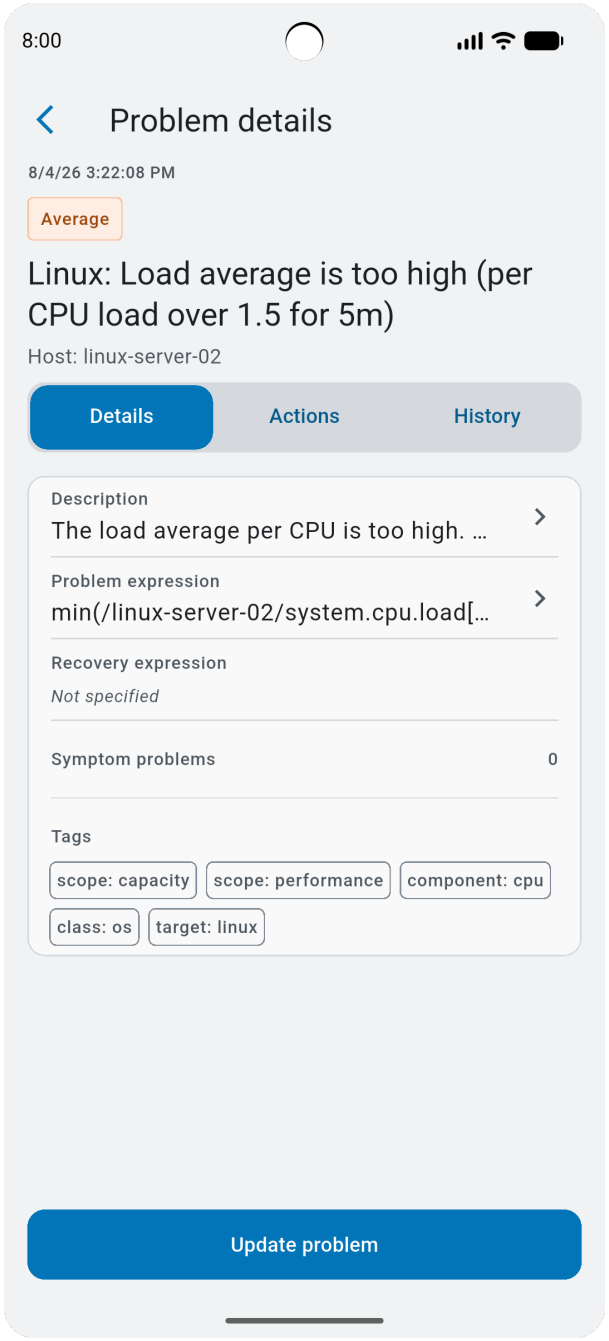
Нажмите меню с тремя точками, чтобы быстро подтвердить проблему или подавить ее на выбранный период.



Сведения о проблеме На экране *Сведения о проблеме* отображаются сведения о проблеме, действия и история.

- Вкладка *Сведения* — отображает описание проблемы, выражения, проблемы-симптомы (нажмите, чтобы просмотреть список проблем) и теги.
- Вкладка *Действия* — отображает временную шкалу проблемы: когда она началась или была подтверждена, было ли отправлено уведомление и другие действия.
- Вкладка *История* — отображает возраст проблемы, ее длительность и время восстановления (или *Недоступно*, если проблема все еще актуальна).

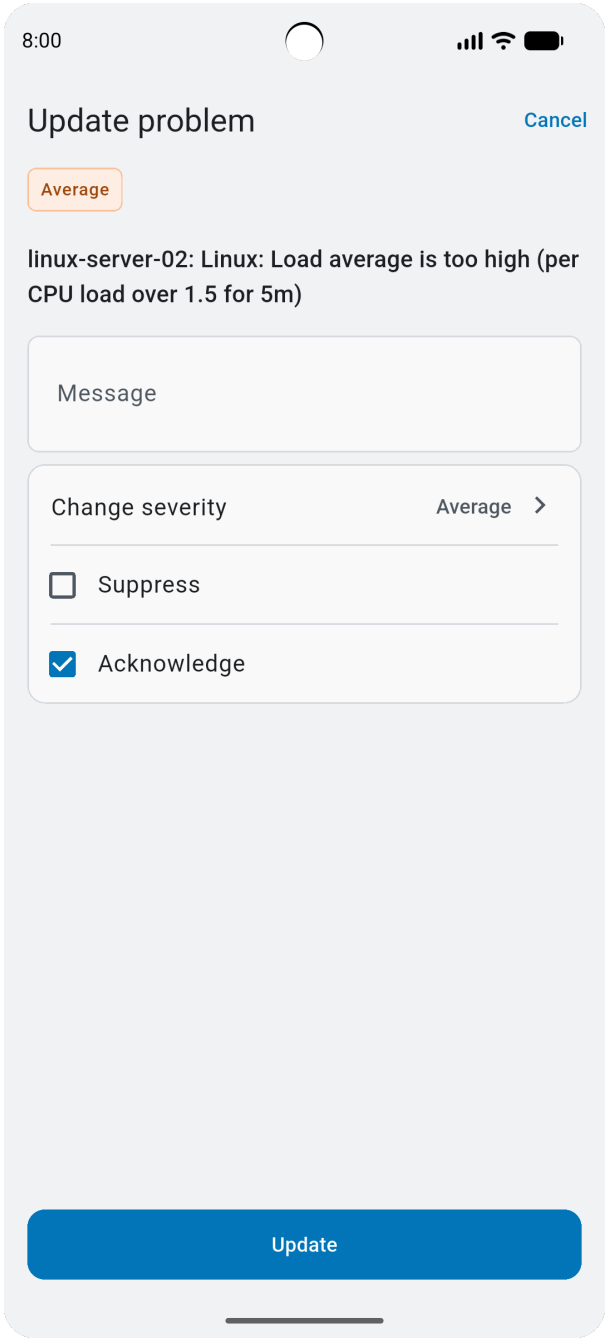
Нажмите *Обновить проблему*, чтобы обновить проблему.



Обновление проблемы Экран *Обновление проблемы* позволяет обновить проблему:

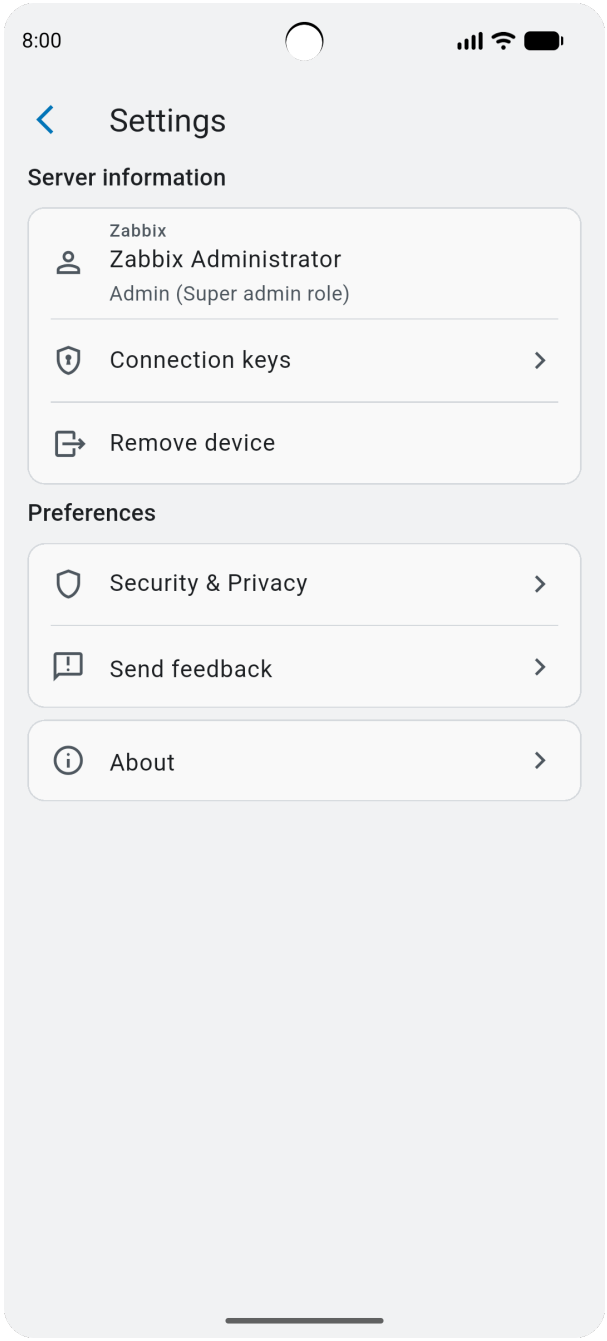
- Добавить сообщение.
- Изменить серьезность проблемы.
- Подавить проблему на выбранный вами период.
- Подтвердить или отменить подтверждение проблемы.
- Закрыть проблему.

Нажмите *Обновить*, чтобы сохранить изменения.



Настройки Экран *Настройки* позволяет получить доступ к настройкам приложения. Он содержит:

- Имя сервера Zabbix (или *Zabbix*, если оно не было настроено при установке веб-интерфейса Zabbix).
- Имя, тип и роль пользователя, к которому добавлено мобильное устройство.
- *Ключи подключения* - откройте, чтобы просмотреть и скопировать идентификатор сервера, идентификатор устройства и другие сведения.
- *Удалить устройство* - нажмите, чтобы удалить мобильное устройство из Zabbix.
- *Безопасность и конфиденциальность* - откройте, чтобы включить блокировку приложения и управлять передачей диагностических данных.
- *Отправить отзыв* - откройте, чтобы отправить отзыв в Zabbix.
- *О приложении* - откройте, чтобы просмотреть версию приложения и другие сведения о нем.



Безопасный мобильный доступ

На этой странице приведены рекомендации по обеспечению безопасности мобильного приложения Zabbix и адаптера Zabbix bridge.

Защита мобильных устройств Чтобы другие люди не могли открыть приложение на вашем мобильном устройстве, включите блокировку приложения в разделе *Настройки*.

Если устройство было утеряно или скомпрометировано, удалите его в веб-интерфейсе Zabbix. Это отзывает и удаляет его учетные данные доступа, поэтому устройство больше не сможет получить доступ к Zabbix. В зависимости от разрешений, предусмотренных вашей ролью пользователя, устройства можно удалять в следующих разделах:

- *Настройки пользователя > Устройства* - любой пользователь может удалить собственные устройства (требуется разрешение *Управление собственными устройствами*).
- *Пользователи > Устройства* - *Суперадминистратор* может удалить устройства любого пользователя (требуется разрешение *Управление устройствами пользователей*).

Также можно ограничить доступ с мобильного устройства к определенным разделам Zabbix. Для этого назначьте пользователям роль пользователя, которая предоставляет только разрешения, необходимые приложению. Минимальный

набор разрешений выглядит следующим образом:

Раздел	Конфигурация
Доступ к элементам интерфейса	Снимите все флажки.
Доступ к сервисам	Установите для параметров *Доступ к сервисам с правом чтения и записи* и *Доступ к сервисам только для чтения* значение `None`.
Доступ к модулям	Снимите все флажки.
Доступ к API	Снимите флажок *Включено*.
Доступ к действиям	Снимите все флажки. Чтобы пользователи могли обновлять проблемы в приложении, также отметьте необходимые им действия: - *Добавление комментариев к проблемам* - *Изменение важности* - *Подтверждение проблем* - *Подавление проблем* - *Закрытие проблем* - *Создание и редактирование собственных средств оповещения* Обратите внимание, что проблемы, которые пользователи могут видеть в приложении, также зависят от их доступа к узлам сети.
Устройства	Установите флажок *Включено*. Чтобы пользователи могли добавлять и удалять собственные мобильные устройства, также установите флажок *Управление собственными устройствами*; в противном случае *Суперадминистратор* должен управлять их устройствами в разделе *Пользователи > Устройства*.

Защита адаптера моста Zabbix Адаптер моста Zabbix хранит свои данные (файл состояния и ключи шифрования) в каталоге, указанном в параметре `DataDir` файла конфигурации адаптера моста Zabbix. Убедитесь, что для каталога установлены права 0700, а для всех хранящихся в нем файлов — права 0600.

Адаптер моста Zabbix передает запросы от мобильных устройств к API Zabbix. Чтобы ограничить методы API, которые можно вызывать через адаптер моста Zabbix, используйте его параметр `AllowedAPIMethods`. Например, разрешите только минимальный набор методов, необходимый мобильному приложению, или даже меньше, если вы не хотите, чтобы использовались некоторые функции мобильного приложения (например, не разрешайте `event.acknowledge`, если пользователи мобильного приложения не должны обновлять проблемы).

Центр разработчика

Центр разработчика

Начните разрабатывать собственные расширения для Zabbix с помощью руководств по плагинам, модулям и виджетам.

Сборка на Zabbix

Плагины Обзор того, как разрабатывать и управлять плагинами Zabbix, чтобы расширять функциональность или интегрироваться с внешними системами.

Библиотека Python для Zabbix Ресурсы по использованию официальной библиотеки `zabbix_utils` для взаимодействия с API Zabbix и создания интеграций.

Модули веб-интерфейса Руководства и справочные материалы по созданию пользовательских модулей, которые расширяют или изменяют веб-интерфейс Zabbix в соответствии с конкретными сценариями использования.

Виджеты Разбор структуры и логики виджетов с инструкциями по созданию пользовательских элементов панели мониторинга, адаптированных под ваши потребности.

Создание плагина

Эта страница предназначена для тех, кто заинтересован в разработке собственного плагина (независимого сборщика данных) для агента Zabbix 2.

Пользовательский плагин расширяет функциональность агента Zabbix 2 за пределами предоставляемых встроенных плагинов и загружаемых плагинов.

Каждый плагин представляет собой пакет *Go*, который определяет структуру и реализует один или несколько интерфейсов (*Exporter*, *Configurator*, *Runner*). Дополнительную информацию см. в разделах Интерфейсы плагинов и Схема взаимодействия.

Это руководство поможет вам создать пользовательский загружаемый плагин.

Дополнительные рекомендации также доступны в следующих репозиториях:

- Пример плагина
- Существующие загружаемые плагины

Что вы создадите Это пошаговое руководство по созданию простого загружаемого плагина с именем **MyIP**. Этот плагин будет реализовывать один ключ элемента данных (`myip`), возвращающий внешний IP-адрес узла сети агента Zabbix.

Шаг 1: Настройка

1. Плагин представляет собой стандартный модуль Go. Начните с инициализации файла `go.mod` в каталоге плагина, чтобы отслеживать зависимости плагина:

```
cd path/to/plugins/myip #
go mod init myip
```

2. Установите обязательную зависимость Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Замените `$LATEST_COMMIT_HASH` на хеш последнего коммита `HEAD` из репозитория `golang.zabbix.com/sdk` в соответствующей ветке релиза. Например:

```
go get golang.zabbix.com/sdk@af85407
```

Обратите внимание, что управление версиями для `golang.zabbix.com/sdk` в настоящее время не поддерживается, но это может измениться в будущем.

Дополнительные зависимости при необходимости можно установить с помощью `go get`.

3. Создайте пустой файл `main.go` для исходного кода плагина:

```
touch main.go
```

Теперь начальная настройка завершена, и плагин готов к разработке.

Шаг 2: Структура плагина Модуль `golang.zabbix.com/sdk`, установленный на предыдущем шаге, предоставляет необходимую платформу для разработки плагинов и обеспечивает единообразную структуру всех плагинов.

1. Настройте базовый поток выполнения.

Начните с определения основного потока выполнения плагина. Добавьте следующий код в `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

Это задает базовый поток выполнения плагина. Позднее функция run будет содержать основную логику плагина.

2. Изучите интерфейсы плагина.

Плагин Zabbix agent 2 должен быть представлен структурой, реализующей интерфейсы из пакета `golang.zabbix.com/sdk/plugin`

- *Accessor* - определяет основные методы, которые должны реализовывать все плагины, например задание имени плагина и обработку тайм-аутов ключей элементов данных.
- Один или несколько следующих функциональных интерфейсов плагина:
  - *Exporter* - выполняет опрос и возвращает значение (или значения), ничего либо ошибку; часто используется вместе с интерфейсом *Collector*.
  - *Collector* - управляет периодическим сбором данных.
  - *Runner* - определяет процедуры запуска и остановки плагина.
  - *Watcher* - обеспечивает независимый опрос метрик в обход внутреннего планировщика агента; полезен для мониторинга на основе ловушек или событий.
  - *Configurator* - определяет способ чтения и применения параметров конфигурации плагина.

Вы можете самостоятельно реализовать эти интерфейсы или использовать реализации по умолчанию, предоставляемые Zabbix Go SDK, изменив их при необходимости. В этом руководстве используются реализации по умолчанию.

3. Создайте структуру плагина.

Теперь импортируйте пакет *plugin* и создайте структуру `myIP`, встраивающую структуру `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

В настоящее время структура myIP удовлетворяет интерфейсу Accessor. Метод для реализации одного из функциональных интерфейсов плагина, *Exporter*, будет добавлен позднее в этом руководстве.

Шаг 3: Определите ключи элементов данных Вашему плагину нужны ключи элементов данных для сбора данных и их передачи на сервер Zabbix или прокси.

1. Импортируйте пакет *errs* для обработки ошибок:

```
import "golang.zabbix.com/sdk/errs"
```

2. Зарегистрируйте ключи элементов данных с помощью функции `plugin.RegisterMetrics()` внутри функции `run()`:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Чтобы зарегистрировать несколько ключей элементов данных, повторите параметры *имя метрики* и *описание* для каждой метрики.
Например:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Шаг 4: Настройка обработчика Обработчик обеспечивает взаимодействие между агентом и плагином.

1. Импортируйте пакет *container*:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Внутри функции `run()` добавьте код для создания и настройки обработчика:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Шаг 5: Реализуйте сбор данных Сбор данных выполняется через интерфейс Exporter, который описывает метод Export:

```
func Export(
  key string,             //                    .
  params []string,        //      ,                      (`myip[arg1, arg2]`).
  context ContextProvider //                              .
) (any, error)
```

1. Импортируйте необходимые пакеты для HTTP-запросов и чтения ответа:

```
import (
    "io"
    "net/http"
)
```

2. Реализуйте метод Export для структуры myIP:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    //                                                    `key`.
    //                    ,          `key`          .
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }
```

```
    //                        agent 2.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    //                  .

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Шаг 6: Сборка и настройка плагина

1. Чтобы собрать плагин, выполните:

```
go mod tidy
go build
```

Это должно создать исполняемый файл `myip` в текущем каталоге.

2. Настройте агент Zabbix 2 для использования плагина:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Замените `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` на путь к `myip`, созданному на шаге 5.

Имя плагина в имени параметра конфигурации (*MyIP* в этом руководстве) должно совпадать с именем плагина, определенным в функции *plugin.RegisterMetrics()*.

3. Чтобы проверить плагин и его элемент данных `myip`, выполните:

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

Вывод должен содержать внешний IP-адрес вашего узла сети и выглядеть примерно так:

```
myip                                            [s|192.0.2.1]
```

На этом вы создали простой загружаемый плагин для агента Zabbix 2. Поздравляем!

```
package main

import (
    "io"
    "net/http"

    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
```

```
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    p := &myIP{}

    //                      `myip`.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             //
        "myip",                             //
        "Returns the host's IP address.", //
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    //              .
    h, err := container.NewHandler("MyIP") //
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    //                                                  .
    //          p.Logger.Infof, p.Logger.Debugf    . .
    p.Logger = h

    //                 .
    //                                          .
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    //                                                        `key`.
    //                    ,             `key`         .
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    //                           2.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    //                   .
```

```
    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Полный исходный код

Интерфейсы плагинов

В этом разделе описаны доступные интерфейсы плагинов.

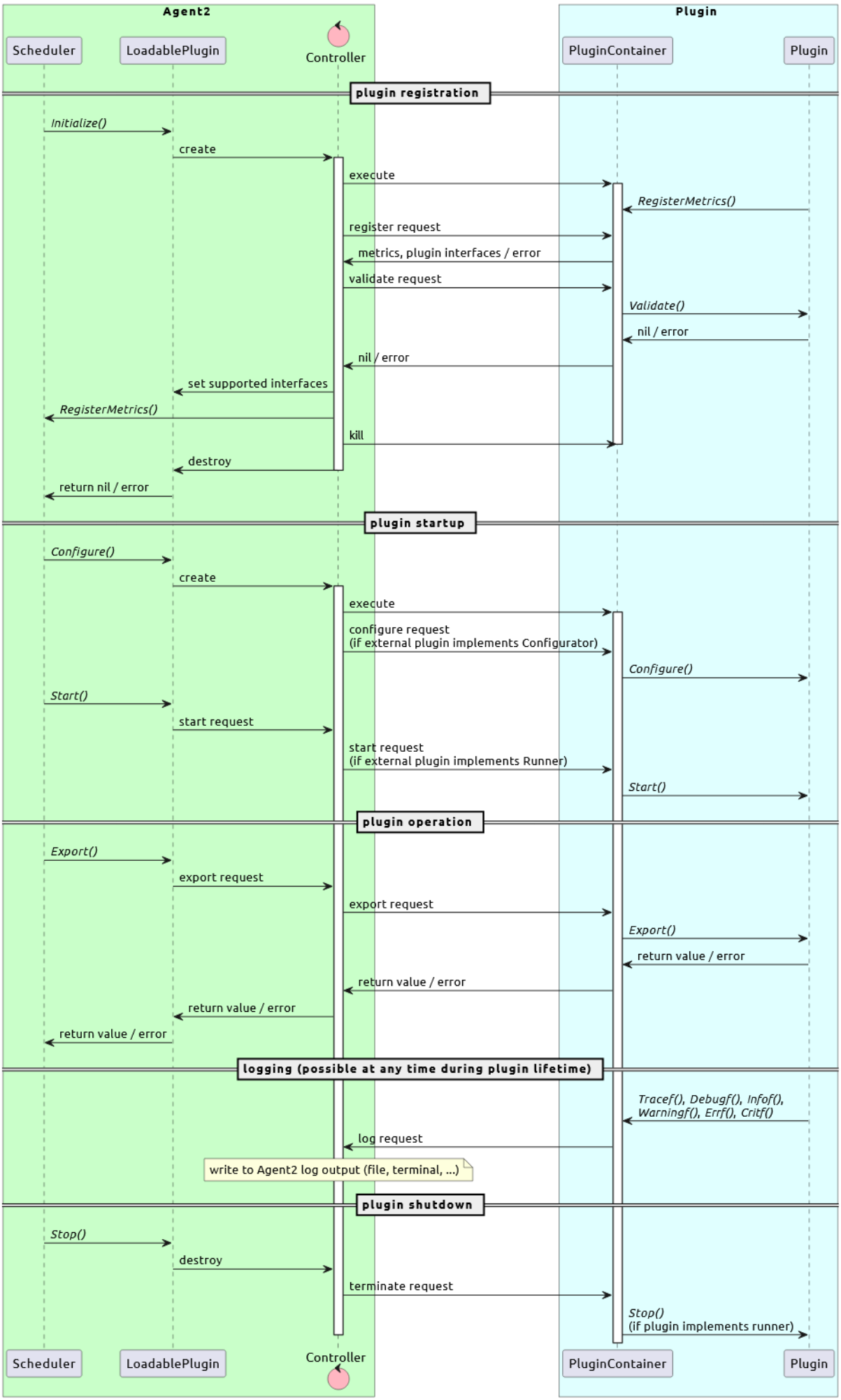
plugin.Exporter *Exporter* - это простейший интерфейс, который выполняет опрос и возвращает значение (значения), ничего или ошибку. Он принимает подготовленный ключ элемента, параметры и контекст. Доступ ко всем остальным интерфейсам плагинов является эксклюзивным, и ни один метод не может быть вызван, если плагин уже выполняет задачу. Кроме того, существует ограничение в 100 одновременных вызовов *Export()* на каждый плагин, которое можно уменьшить в соответствии с требованиями каждого плагина.

plugin.Configurator Интерфейс *Configurator* предоставляет параметры конфигурации плагина из файлов конфигурации Zabbix агента 2.

plugin.Runner Интерфейс *Runner* предоставляет средства для выполнения инициализации при запуске (активации) плагина и деинициализации при остановке (деактивации) плагина. Например, плагин может запускать/останавливать некоторую фоновую *goroutine*, реализуя интерфейс Runner.

Схема подключения Zabbix агент 2 устанавливает двунаправленное соединение с плагинами, используя UNIX-сокеты в Linux и именованные каналы в Windows.

Приведенная ниже схема подключения иллюстрирует процесс взаимодействия между Zabbix агентом 2 и загружаемым плагином, а также процесс сбора метрик.


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

Библиотека Python для Zabbix

zabbix_utils — это официальная библиотека Python для Zabbix.

С помощью zabbix_utils вы можете:

- Использовать API Zabbix для управления объектами Zabbix (создавать узлы сети, обновлять элементы данных, получать события и т. д.).
- Собирать данные с агента Zabbix (аналогично Zabbix get).
- Отправлять данные на сервер Zabbix или прокси (аналогично Zabbix sender).

Библиотека поддерживает как синхронное, так и асинхронное выполнение скриптов, что позволяет использовать ее как для простых рабочих процессов, так и для крупномасштабных параллельных операций, таких как массовый сбор данных, пакетный экспорт, обработка оповещений и эффективное масштабирование системы.

Библиотека совместима с поддерживаемыми версиями сервера Zabbix и современными выпусками Python.

Чтобы начать работу, установите библиотеку, а затем следуйте руководству quickstart.

Установка

zabbix_utils можно установить любым из способов, описанных ниже.

Требования:

- Zabbix 6.0 или новее (проверено на 6.0, 7.0, 7.2, 7.4)
- Python 3.8 или новее (проверено на 3.8–3.13)
- aiohttp — асинхронный HTTP-фреймворк (требуется только для асинхронного режима)

Python Package Index (PyPI)

Это наиболее распространенный способ для большинства сред:

```
pip install zabbix_utils
pip install zabbix_utils[async] #             async
```

Репозиторий Zabbix

Используйте этот способ, если вы предпочитаете управлять зависимостями через менеджер пакетов вашей системы.

Скачайте пакеты Zabbix для вашего дистрибутива Linux, а затем выполните следующие команды.

Для RHEL и производных:

```
dnf install python3-zabbix-utils
dnf install epel-release #             async
dnf install python3-aiohttp #             async
```

Для Debian/Ubuntu и производных:

```
apt install python3-zabbix-utils
apt install python3-aiohttp #             async
```

Исходный код (GitHub)

Используйте этот метод, если вы предпочитаете последнюю версию разработки:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt #             async
python3 setup.py install
```

Чтобы установить зависимости, необходимые для асинхронного режима, вы также можете использовать один из методов, описанных выше.

После того как вы установили zabbix_utils, вы можете использовать его в своем скрипте.

Это руководство по быстрому началу работы показывает, как:

- Получить имя узла сети из агента Zabbix.
- Создать узел сети и элемент данных типа trapper с помощью API Zabbix.
- Отправить значение в элемент данных.

В руководстве скрипт приводится пошагово, и каждая часть объясняется по мере ее появления. Полный скрипт приведен в конце страницы.

Также предполагается, что ваш сервер Zabbix, агент и API работают локально.

Получение имени узла сети из агента Zabbix

Начните с получения имени узла сети, на котором запущен агент Zabbix.
Это имя узла сети понадобится вам для создания узла сети с помощью API Zabbix.

1. Импортируйте класс `Getter` из `zabbix_utils`.
Этот класс работает как Zabbix get и позволяет запрашивать данные у агента Zabbix.

2. Создайте экземпляр `Getter`, который подключается к локальному агенту Zabbix по адресу `127.0.0.1` на порту 10050.

3. Вызовите метод `get()` у экземпляра `Getter`, чтобы запросить имя узла сети у агента Zabbix.

- Метод `get()` принимает в качестве параметра ключ элемента данных и отправляет запрос агенту Zabbix для этого элемента данных.
- Метод `get()` возвращает объект, а имя узла сети хранится в атрибуте `value` этого объекта.
- `system.hostname` — это встроенный ключ агента Zabbix, который возвращает имя узла сети.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Подключение и вход в Zabbix API

Далее подключитесь к Zabbix API. Это позволит вашему скрипту управлять объектами Zabbix, такими как узлы сети и элементы данных.

1. Импортируйте класс `ZabbixAPI` из zabbix_utils.

2. Создайте экземпляр `ZabbixAPI` и укажите URL вашего веб-интерфейса Zabbix.

3. Вызовите метод `login()` для экземпляра `ZabbixAPI` и укажите имя пользователя и пароль. У учетной записи должно быть разрешение на доступ к Zabbix API.

4. Вызовите метод `logout()`, чтобы закрыть сеанс после выполнения операций API.

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### API operations go here

api.logout()
```

Создание узла сети в Zabbix

Теперь, когда у вас есть имя узла сети из агента Zabbix и вы подключены к API Zabbix, вы можете создать новый узел сети в Zabbix.

1. Вызовите API-метод `host.create()` для экземпляра `ZabbixAPI` и укажите сведения об узле сети:

- `host` - задайте значение переменной `hostname`, содержащей имя узла сети, которое вы получили из агента Zabbix.
- `interfaces` - содержит сведения о подключении для агента Zabbix, который работает на узле сети.
- `groups` - содержит как минимум одну группу узлов сети, к которой должен принадлежать узел сети.

2. Поскольку API Zabbix возвращает ID только что созданного узла сети, сохраните этот ID в переменной `host_id` для последующего использования.

3. Выведите сообщение для подтверждения создания узла сети.

```
api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")
```

Создание нового элемента данных в Zabbix

После создания узла сети вы можете добавить к нему элемент данных.

1. Определите уникальный ключ для создаваемого элемента данных.

2. Вызовите метод `item.create()` для экземпляра `ZabbixAPI` и передайте сведения об элементе данных:

- `hostid` - установите значение переменной `host_id`, содержащей ID созданного вами узла сети.
- `name` - имя элемента данных.
- `key_` - установите значение переменной `item_key`, которую вы только что определили.
- `type` - установите значение 2 (элемент trapper), требуется для получения значений, отправляемых из вашего скрипта на следующих шагах.
- `value_type` - установите значение 3 (числовой без знака), тип данных, которые хранит этот элемент данных.

3. Поскольку Zabbix API возвращает ID вновь созданного элемента данных, сохраните этот ID в переменной `item_id` для последующего использования.

4. Выведите сообщение для подтверждения того, что элемент данных был создан.

```
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3,
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")
```

Отправка значения на узел сети

Теперь, когда вы создали узел сети и элемент данных, вы можете отправить на него данные.

1. Импортируйте класс `time` и подождите несколько секунд перед отправкой данных.
Это гарантирует, что Zabbix полностью обработал ваш новый узел сети и элемент данных.

2. Импортируйте класс `Sender` из `zabbix_utils`.
Этот класс работает как Zabbix sender и позволяет вашему скрипту отправлять данные в Zabbix.

3. Создайте экземпляр `Sender`, который подключается к локальному серверу Zabbix по адресу `127.0.0.1` на порту `10051`.

4. Вызовите метод `send_value()` у экземпляра `Sender`, чтобы отправить значение в элемент данных вашего узла сети, и укажите следующие данные:

- `hostname` - установите значение переменной `hostname`, содержащей имя узла сети, полученное от агента Zabbix.
- `item_key` - установите значение переменной `item_key`, которую вы только что определили ранее в скрипте.
- `1` - значение для отправки.

4. Выведите сообщение для подтверждения того, что значение было отправлено.

```
import time
time.sleep(10)

from zabbix_utils import Sender

sender = Sender(server='127.0.0.1', port=10051)

response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

После успешного выполнения этого кода вы должны увидеть сообщение об успешной отправке, указывающее, что Zabbix получил ваше значение:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Теперь вы также можете проверить свой веб-интерфейс Zabbix (*Monitoring* > *Latest data*) и увидеть это значение.

Полный скрипт

Ниже приведен полный скрипт, который объединяет все шаги: получение имени узла сети, создание узла сети и элемента данных с использованием Zabbix API, а также отправку данных в Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

###                      Zabbix
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

###                    Zabbix API
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

###              Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"       '{hostname}'       ID {host_id}")

###                 Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
```

```
)

item_id = item['itemids'][0]

print(f"         '{item_key}'       ID {item_id}")

###       API
api.logout()

###      ,     Zabbix
time.sleep(10)

###
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"    Sender: {response}")
```

Использование Zabbix API

zabbix_utils позволяет использовать Zabbix API для управления объектами Zabbix, включая создание узлов сети, обновление элементов данных, получение событий и многое другое.

Запросы API можно выполнять в синхронном или асинхронном режиме:

- В синхронном режиме ваш скрипт Python отправляет запрос и ожидает ответа, прежде чем продолжить выполнение, что подходит для простых, последовательных и предсказуемых операций.
- В асинхронном режиме скрипт отправляет запросы, не ожидая ответа на каждый из них, что позволяет выполнять другие операции параллельно; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует похожие подходы. Дополнительные примеры доступны в репозитории GitHub zabbix_utils.

Импорт

Чтобы использовать zabbix_utils для работы с Zabbix API, импортируйте класс ZabbixAPI в свой скрипт:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Если вы уже используете библиотеку Python от сообщества, обычно можно заменить этот импорт на ZabbixAPI из zabbix_utils.

Вход в систему

Перед выполнением API-запросов необходимо создать экземпляр ZabbixAPI и выполнить вход в Zabbix API.

Выберите метод, который лучше всего соответствует тому, как вы управляете учетными данными. Вы можете использовать имя пользователя и пароль или API tokens.

Во время инициализации

При создании экземпляра ZabbixAPI вы можете сразу указать URL веб-интерфейса Zabbix и свои учетные данные:

```
###                :
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

###     API:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Если вы предпочитаете сгруппировать параметры подключения, вы можете передать их в виде словаря Python и распаковать его с помощью синтаксиса именованных аргументов Python:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
```

```
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Использование `login()`

Сначала можно создать экземпляр ZabbixAPI, указав только URL веб-интерфейса Zabbix, а затем использовать метод `login()` с вашими учетными данными:

```
###                 :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API token:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Использование переменных среды

Сначала вы можете сохранить свои учетные данные в переменных среды, например, с помощью терминала:

```
###                 :
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

###    :
export ZABBIX_TOKEN="your_api_token"
```

Затем, если вы запустите свой скрипт из той же сессии терминала, экземпляру ZabbixAPI в скрипте потребуется только URL веб-интерфейса Zabbix:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Вы также можете сохранить URL в переменной среды:

```
###                 :
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

###    :
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

Когда URL и учетные данные заданы как переменные среды, экземпляру ZabbixAPI в скрипте не требуется вообще никаких аргументов:

```
api = ZabbixAPI()
```

Выход из системы

Если вы вошли в систему, используя имя пользователя и пароль, вызовите метод `logout()` после завершения операций с API:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API actions go here

api.logout()
```

> **Note:**
> При использовании токенов API вызывать метод `logout()` не требуется.

API-запросы

После входа в систему вы можете выполнять любые API-запросы, вызывая методы, описанные в справочнике методов Zabbix API.

Методы API вызываются в следующем формате:

```
api_instance.zabbix_object.method(parameters)
```

> **Note:**
> Некоторые имена методов или объектов Zabbix API используют слова, которые являются зарезервированными ключевыми словами в Python (например, `import`). Чтобы избежать ошибок Python, добавляйте символ подчеркивания к имени метода или объекта при использовании его в Python (например, `api.configuration.import_`).

Например, чтобы получить список узлов сети с помощью `host.get`:

```
### 1.         ZabbixAPI    zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2.             ZabbixAPI           :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3.            ,               :
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4.                                        :
for host in hosts:
    print(host)

### 5.        API,             :
api.logout()
```

Асинхронный режим

Асинхронный режим позволяет вашему скрипту отправлять API-запросы, не дожидаясь завершения каждого из них. Это может сделать ваш скрипт более эффективным, когда ему нужно выполнять много API-запросов или когда некоторые запросы занимают много времени.

При использовании асинхронного режима есть важные отличия по сравнению с синхронным режимом:

- Импортируйте модуль Python `asyncio` (сначала необходимо установить требуемые зависимости).
- Импортируйте `AsyncZabbixAPI` вместо `ZabbixAPI`.
- Пишите код внутри функции `async`.
- Используйте `await` при вызове методов API, включая `login()` и `logout()`.

> **Attention:**
> При создании экземпляра `AsyncZabbixAPI` вы не можете выполнить вход во время инициализации.

Например, чтобы получить список узлов сети с помощью `host.get` в асинхронном режиме:

```
### 1.         asyncio                  AsyncZabbixAPI    zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2.                          ,                           API:
async def main():

    # 3.             AsyncZabbixAPI            (    await):
    api = AsyncZabbixAPI(url="127.0.0.1")
```

```
    await api.login(user="User", password="zabbix")

    # 4.            ,               (    await):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5.                                        :
    for host in hosts:
        print(host)

    # 6.             API,             (    await):
    await api.logout()

### 7.                  main()               asyncio:
asyncio.run(main())
```

Примеры

Следующие примеры показывают распространенные задачи Zabbix API с использованием библиотеки zabbix_utils.

Дополнительные примеры доступны в каталоге `examples` в репозитории GitHub zabbix_utils.

Сбор данных с агента Zabbix

Обзор

zabbix_utils позволяет собирать данные с агента Zabbix (аналогично Zabbix get).

Данные можно собирать в синхронном или асинхронном режиме:

- В синхронном режиме ваш Python-скрипт запрашивает данные и ожидает их получения, прежде чем продолжить выполнение, что подходит для простых, последовательных и предсказуемых операций.
- В асинхронном режиме скрипт запрашивает данные, не ожидая каждого ответа, что позволяет выполнять другие операции параллельно; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует похожие подходы. Дополнительные примеры доступны в репозитории GitHub zabbix_utils.

Импорт

Чтобы использовать zabbix_utils для сбора значений элементов данных, импортируйте класс `Getter` в ваш скрипт Python:

```
from zabbix_utils import Getter
```

Данные запроса

Чтобы запросить значение элемента данных:

1. Создайте экземпляр `Getter`, указав IP-адрес и порт вашего агента Zabbix.
2. Вызовите метод `get()` у экземпляра `Getter`, указав ключ элемента данных, который нужно получить.

Например, чтобы запросить данные для элемента данных `system.uname`:

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Использование IP-адреса, отличного от значения по умолчанию

Если сервер, на котором выполняется ваш скрипт, имеет несколько IP-адресов, вы можете указать `source_ip`, который `Getter` будет использовать при подключении к агенту Zabbix:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Использование тайм-аута

Вы можете задать `timeout` ожидания ответа для `Getter`, чтобы контролировать, как долго ваш скрипт должен ждать ответа от агента Zabbix, прежде чем прекратить ожидание:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Использование шифрования

`Getter` не включает встроенную поддержку шифрования, но вы можете добавить ее, создав обертку с использованием сторонних библиотек:

```
def psk_wrapper(sock, tls):
    # ...
    #            TLS PSK
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Ответ

Ответ от агента Zabbix обрабатывается библиотекой и возвращается как объект `AgentResponse`:

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

Асинхронный режим

Асинхронный режим позволяет вашему скрипту собирать значения, не дожидаясь получения каждого из них. Это может сделать ваш скрипт более эффективным, когда ему нужно собрать много значений или когда получение некоторых значений занимает много времени.

При использовании асинхронного режима есть важные отличия по сравнению с синхронным режимом:

- Импортируйте модуль Python `asyncio` (сначала необходимо установить требуемые зависимости).
- Импортируйте `AsyncGetter` вместо `Getter`.
- Пишите код внутри асинхронной функции `async`.
- Используйте `await` при вызове метода `get()`.

Например, чтобы собрать одно значение с использованием асинхронного режима:

```
### 1.        asyncio                AsyncGetter   zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter
```

```
### 2.                          ,                              :
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3.            system.uname        Zabbix (            await):
    response = await agent.get('system.uname')

    # 4.           ,              Zabbix:
    print(response.value)

### 5.                   main()                asyncio:
asyncio.run(main())
```

Отправка данных на сервер Zabbix или прокси

zabbix_utils позволяет отправлять значения элементов данных на элемент данных типа trapper на сервер Zabbix или прокси (аналогично Zabbix sender).

Вы можете отправить одно значение, несколько значений или даже направить данные на несколько кластеров Zabbix.

Данные можно отправлять в синхронном или асинхронном режиме:

- В синхронном режиме ваш скрипт Python отправляет значения и ожидает ответа перед продолжением; это подходит для простых, последовательных и предсказуемых операций.
- В асинхронном режиме скрипт отправляет значения, не ожидая каждого ответа, что позволяет другим операциям выполняться параллельно; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует похожие подходы. Дополнительные примеры доступны в репозитории GitHub zabbix_utils.

Импорт

Чтобы использовать zabbix_utils для отправки значений элементов данных, импортируйте класс `Sender` в свой скрипт:

```
from zabbix_utils import Sender
```

Для отправки нескольких значений вы также можете импортировать класс `ItemValue`:

```
from zabbix_utils import Sender, ItemValue
```

Отправка одного значения

Чтобы отправить значение элемента данных:

1. Создайте экземпляр `Sender`, указав IP-адрес и порт вашего сервера Zabbix или прокси.
2. Вызовите метод `send_value()` для экземпляра `Sender` в следующем формате:

```
sender_instance.send_value('      ', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Например, чтобы отправить 1 в trapper-элемент данных `service.status` на узле сети `Linux server`:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Использование IP-адреса, отличного от значения по умолчанию

Если сервер, на котором выполняется ваш скрипт, имеет несколько IP-адресов, вы можете указать `source_ip`, который `Sender` будет использовать при отправке значений на сервер Zabbix или прокси:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Использование тайм-аута

Вы можете задать `timeout` ответа для `Sender`, чтобы управлять тем, как долго ваш скрипт должен ждать ответа от сервера Zabbix или прокси, прежде чем прекратить ожидание:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Использование файла конфигурации агента

Вы можете разрешить zabbix_utils считывать параметры Server или ServerActive из локального файла конфигурации Zabbix агента или агента 2. В этом случае при создании экземпляра Sender не нужно указывать параметры подключения:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Если параметр ServerActive содержит один или несколько кластеров Zabbix с несколькими экземплярами сервера, Sender отправляет данные на первый доступный сервер в каждом кластере. Если параметр ServerActive не задан, используется адрес из Server с портом по умолчанию (10051).

Использование шифрования

Sender не включает встроенную поддержку шифрования, но вы можете добавить ее, создав обертку с использованием сторонних библиотек:

```
def psk_wrapper(sock, tls):
    # ...
    #             TLS PSK
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Ответ для одного значения

Ответ, возвращаемый сервером или прокси Zabbix, обрабатывается библиотекой и возвращается в виде объекта TrapperResponse:

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

Асинхронный режим

Асинхронный режим позволяет вашему скрипту Python отправлять значения, не дожидаясь ответа от сервера Zabbix или прокси. Это может сделать ваш скрипт более эффективным, когда ему нужно отправлять много значений или когда отправка некоторых значений занимает много времени.

При использовании асинхронного режима есть несколько важных отличий по сравнению с синхронным режимом:

- Импортируйте модуль Python asyncio (сначала необходимо установить требуемые зависимости).
- Импортируйте AsyncSender вместо Sender.
- Пишите код внутри функции async.
- Используйте await при вызове метода send_value().

Например, чтобы отправить одно значение с использованием асинхронного режима:

```
### 1.        asyncio                AsyncSender   zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2.                        ,                                    (               await):
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3.         ,             Zabbix       :
    print(response)

### 4.                  main()               asyncio:
asyncio.run(main())
```

Отправка нескольких значений

Чтобы отправить несколько значений:

1. Подготовьте массив объектов `ItemValue`, каждый из которых использует тот же формат, что и метод `send_value()`.
2. Создайте экземпляр `Sender`, указав IP-адрес и порт вашего сервера Zabbix или прокси.
3. Вызовите метод `send()` (вместо `send_value()`) у экземпляра `Sender`, указав массив объектов со значениями для отправки.

Например, чтобы отправить пять значений на разные узлы сети:

```
items = [
    ItemValue('server-de', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-fr', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-uk', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-nl', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-pl', 'service.status', 'up', 1770887205, 100),
]

sender = Sender(server='127.0.0.1', port=10051)
response = sender.send(items)
```

Использование пользовательского размера фрагмента

Если вам нужно отправить больше значений, чем элемент данных типа trapper может принять за один запрос, вы можете разделить их на фрагменты.

По умолчанию размер фрагмента составляет 250 значений. Вы можете изменить его, задав параметр `chunk_size` при создании экземпляра `Sender`.

Например, чтобы отправить пять значений в трех фрагментах (2-2-1), задайте параметр `chunk_size` равным 2:

```
items = [
    ItemValue('server-de', 'service.status', 'up'),
    ItemValue('server-fr', 'service.status', 'up'),
    ItemValue('server-uk', 'service.status', 'up'),
    ItemValue('server-nl', 'service.status', 'up'),
    ItemValue('server-pl', 'service.status', 'up'),
]

sender = Sender(server='127.0.0.1', port=10051, chunk_size=2)
response = sender.send(items)
```

Отправка значений в несколько кластеров Zabbix

Чтобы отправить значения в несколько кластеров Zabbix:

1. Подготовьте массив кластеров Zabbix. Если у кластера несколько узлов, значение будет отправлено на первый **доступный** узел каждого кластера.
2. Создайте `Sender`, указав ваш массив кластеров Zabbix.
3. Вызовите метод `send_value()` для экземпляра `Sender`, используя тот же формат, что и метод `send_value()`.

Например, чтобы отправить значение на первый доступный узел в каждом кластере:

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

Ответ для нескольких значений

По умолчанию Sender возвращает агрегированный результат отправки значений по всем узлам сети или кластерам:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Если вам нужна более подробная информация, вы можете просмотреть результаты для каждого кластера и каждого фрагмента с помощью атрибута `response.details`:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"                  {response.time}")
else:
    print(response.details)
    # {
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"      {resp.processed}   {resp.total}   {node.address}:{node.port}")
            #       1   1   127.0.0.1:10051
            #       1   1   zabbix.example.local:10051
```

Отладочное логирование

Чтобы устранить неполадки при работе с Zabbix API, сборе данных с агента Zabbix или отправке данных на сервер Zabbix или прокси, можно включить отладочное логирование.

Библиотека zabbix_utils использует стандартный модуль Python `logging`.

Чтобы видеть подробные отладочные сообщения из библиотеки:

1. Импортируйте модуль Python `logging` в свой скрипт.
2. Настройте систему логирования так, чтобы сообщения выводились в удобочитаемом формате, и установите уровень логирования DEBUG.

```
import logging
from zabbix_utils import Getter

logging.basicConfig(
    format=u'[%(asctime)s] %(levelname)s %(message)s',
    level=logging.DEBUG
)

agent = Getter(host='127.0.0.1', port=10050)
resp = agent.get('system.uname')

print(resp.value)
```

При включенном отладочном логировании подробная информация о запросах и ответах выводится, например, так:

```
[2023-10-01 12:00:01,587] DEBUG Content of the packet: b'ZBXD\x01\x0c\x00\x00\x00\x00\x00\x00\x00system.un
[2023-10-01 12:00:01,722] DEBUG Zabbix response header: b'ZBXD\x01C\x00\x00\x00C\x00\x00\x00'
[2023-10-01 12:00:01,723] DEBUG Zabbix response body: Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
[2023-10-01 12:00:01,724] DEBUG Response from [127.0.0.1:10050]: Linux test_server 5.15.0-3.60.5.1.el9uek.
Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
```

Модули

Что такое модуль PHP веб-интерфейса?

- Модуль - это объект с уникальным идентификатором, именем, описанием, автором и другими полями, определенными в его файле манифеста, а также PHP, Javascript и другими файлами, расположенными в одном каталоге внутри подкаталога *modules* каталога установки веб-интерфейса Zabbix (например, *zabbix/ui/modules*).
- Модуль должен соответствовать простым правилам, чтобы гарантировать правильную работу.
- Модуль должен быть установлен (распакован) и включен в веб-интерфейсе администратором.

Для чего можно использовать модуль

- Добавление нового функционала через пользовательские разделы веб-интерфейса;
- Создание пользовательских типов виджетов панели (см. модули виджетов);
- Переопределение или расширение существующего функционала.

Для чего нельзя использовать модуль

- Регистрация нового метода API или изменение существующего.

Как работают модули

- Включенный модуль запускается при каждом HTTP-запросе перед выполнением кода действия.
- Модуль зарегистрирует новые действия или переопределит существующие.
- Модуль добавит новые разделы интерфейса и удалит или переопределит существующие.
- При необходимости модуль будет подключаться к событиям внешнего интерфейса, таким как onBeforeAction и on-Terminate.
- Запрошенное действие выполняется путем запуска кода действия - либо кода по умолчанию, либо определенного в модуле.

Куда идти дальше Независимо от того, предпочитаете ли вы учиться на практике или сначала прочитать рекомендации, эти страницы содержат информацию и шаги, необходимые для создания собственных модулей:

- Пошаговые руководства по написанию вашего первого модуля
- Файловая структура модуля
- Особенности модуля виджетов
- Примеры модулей для повторного использования

Файловая структура модуля

Весь код, связанный с модулем, хранится в одном каталоге внутри каталога **modules** вашей установки веб-интерфейса Zabbix (например, *zabbix/ui/modules*).

```
example_module_directory/          (      )
   manifest.json                   (      )                    .
   Module.php                                                        .
   actions/                                                    .
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                                  .
       example.something.view.php
       example.something.delete.php
   assets/                                                       ,                    .                  n
       js/                                        JavaScript,                 .
           example.something.view.js.php
       css/                                    CSS-   ,                .
           example.something.css
        image.png                                      ,                .
        example.something.file                                            .
```

Дерево файлов модуля

Написание модуля Процесс создания примера модуля состоит из следующих шагов (где доступно, нажмите на имя файла или папки, чтобы просмотреть дополнительные сведения о шаге):

1. Создайте новый каталог для модуля внутри **zabbix/ui/modules/**.
2. Добавьте файл manifest.json с метаданными модуля.
3. Создайте папку *views* и определите представление(я) модуля.
4. Создайте папку *actions* и определите действие(я) модуля.
5. Создайте файл Module.php (или Widget.php для виджетов панели) и определите правила инициализации и обработки событий.
6. Создайте папку *assets* для файлов JavaScript (размещайте их в *assets/js*), CSS-стилей (размещайте их в *assets/css*) или любых других дополнительных файлов.
7. Убедитесь, что в manifest.json указаны необходимые представления, действия и файлы ресурсов.
8. Зарегистрируйте модуль во веб-интерфейсе Zabbix и начните его использовать.

Note:
Вы можете зарегистрировать и включить модуль сразу после создания файла manifest.json. После включения модуля вы сможете сразу просматривать все изменения, внесенные в файлы модуля, просто обновив веб-интерфейс Zabbix.

manifest.json

Любому модулю нужен файл manifest.json. Файл должен находиться в основном каталоге модуля (например, *zabbix/ui/modules/module_name/manifest.json*).

Минимально необходимый набор полей, которые должен содержать manifest.json:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
```

```
    "version": "1.0"
}
```

Параметры, поддерживаемые в **manifest.json** (щелкните имя параметра, чтобы открыть подробное описание):

Parameter	Description	Required
manifest_version	Версия манифеста модуля.	Yes
id	Уникальный ID модуля.	
name	Имя модуля, которое будет отображаться в разделе администрирования.	
namespace	Пространство имен PHP для классов модуля.	
version	Версия модуля.	
type	Тип модуля. Для виджета должно быть установлено значение *widget*	Yes for widgets, otherwise no
widget	Конфигурация виджета. Используется только для виджетов.	
actions	Действия, которые нужно зарегистрировать вместе с модулем.	
assets	CSS-стили и файлы JavaScript для включения.	No
author	Автор модуля.	
config	Значения по умолчанию для пользовательских параметров модуля.	
description	Описание модуля.	
url	Ссылка на описание модуля.	

manifest_version

Версия манифеста модуля. На данный момент поддерживается версия **2.0**. *Тип*: Double

Пример:

```
"manifest_version": 2.0
```

id

Идентификатор модуля. Должен быть уникальным. Чтобы избежать конфликта имён в будущем, рекомендуется использовать для модулей префикс (имя автора или название компании, или любой другой). Например, если модуль является примером для уроков и имя модуля «My module», то идентификатор будет «example_my_module».

Тип: String

Пример:

```
"id": "example_my_module"
```

name

Имя модуля, которое будет отображаться в секторе Администрирование.

Тип: String

Пример

```
"name": "My module"
```

namespace

Пространство имён PHP для классов модулей

Тип: String

Пример:

```
"namespace": "ClockWidget"
```

version

Версия модуля. Версия будет отображаться в секторе Администрирование.

Тип: String

Пример:

```
"version": "1.0"
```

type

Тип модуля. Требуется для виджетов и должен быть равен ”widget”.

Тип: String

По умолчанию: ”module”

Пример:

```
"type": "widget"
```

actions

Действия для регистрации в модуле. Требуется определение ключа объекта *class* для каждого действия, остальные ключи действий не являются обязательными.

Тип: Object

Поддерживаемые ключи объекта, если *type* равно *module*:

- **write.your.action.name** (object) - название действия следует писать строчными буквами [a-z], разделяя слова точками. Поддерживает ключи:
  - **class** (string; обязательный) - имя класса действия.
  - **layout** (string) - макет действия. Поддерживаемые значения: *layout.json*, *layout.htmlpage* (по умолчанию), *null*.
  - **view** (string) - вид действия.

Пример:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

Поддерживаемые ключи объекта, если *type* равно *widget*:

- **widget.{id}.view** (object) - имя файла и класса для просмотра виджета. Замените **{id}** значением id виджета (например, *widget.example_clock.view*). Поддерживает ключи:
  - **class** (string; обязательный) - имя класса действия для режима просмотра виджета для расширения класса CControllerDashboardWidgetView по умолчанию. Исходный файл класса должен находиться в каталоге *actions*.
  - **view** (string) - вид виджета. Должен находиться в каталоге *views*. Если файлом представления является *widget.view.php*, который ожидается по умолчанию, этот параметр можно опустить. Если вы используете другое имя, укажите его здесь.
- **widget.{id}.edit** (object) - имя файла для просмотра конфигурации виджета. Замените **{id}** значением id виджета (например, *widget.example_clock.edit*). Поддерживает ключи:
  - **class** (string; обязательный) - имя класса действия для режима просмотра конфигурации виджета. Исходный файл класса должен находиться в каталоге *actions*.
  - **view** (string) - вид конфигурации виджета. Должен находиться в каталоге *views*. Если файлом представления является *widget.edit.php*, который ожидается по умолчанию, этот параметр можно опустить. Если вы используете другое имя, укажите его здесь.

Пример:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

Стили CSS и файлы JavaScript для включения.

Тип: Object

Поддерживаемые ключи объектов:

- **css** (array) – файлы CSS для включения. Файлы должны находиться в папке *assets/css*.
- **js** (array) — файлы JavaScript для включения. Файлы должны находиться в папке *assets/js*.

Пример:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

author

Автор модуля. Автор будет отображаться в разделе Администрирование.

Тип: String

Пример:

```
"author": "John Smith"
```

config

Значения по умолчанию для опций модуля. Объект может содержать любые пользовательские ключи. Если указано, эти значения будут записаны в базу данных во время регистрации модуля. Новые переменные, добавленные позже, будут записаны при первом вызове. В дальнейшем значения переменных можно будет изменить только непосредственно в базе данных.

Тип: Object

Пример:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

description

Описание модуля.

Тип: String

Пример:

```
"description": "          ."
```

widget

Настройка виджета. Используется, если для *type* установлено значение *widget*.

Тип: Object

Поддерживаемые ключи объекта:

- **имя** (string) — используется в списке виджетов и в качестве заголовка по умолчанию. Если пусто, будет использоваться параметр «name» из модуля.
- **size** (object) – размеры виджета по умолчанию. Поддерживает ключи:
  - *width* (integer) — ширина виджета по умолчанию.
  - *height* (integer) — высота виджета по умолчанию.
- **form_class** (string) - класс с формой полей виджета. Должен находиться в каталоге *includes*. Если классом является *WidgetForm.php*, который ожидается по умолчанию, этот параметр можно опустить. Если вы используете другое имя, укажите его здесь.
- **js_class** (string) — имя класса JavaScript для режима просмотра виджета для расширения класса CWidget по умолчанию. Класс будет загружен с информационной панелью. Исходный файл класса должен находиться в каталоге assets/js. См. также: assets.
- **use_time_selector** (boolean) — определяет, требует ли виджет выбора времени на панели управления. Поддерживаемые значения: *true*, *false* (по умолчанию).
- **refresh_rate** (integer) — частота обновления виджета в секундах (по умолчанию: 60).

Пример:

```
"widget": {
    "name": "",
    "size": {
    "width": 12,
    "height": 5
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

Ссылка на описание модуля. Для виджетов эта ссылка будет открываться при нажатии на значок справки ? в окне *Добавить виджет* или *Изменить виджет*. Если **url** не указан, при нажатии на значок справки будет открываться общая страница Виджеты панелей мониторинга.

Тип: String

Пример:

```
"url": "http://example.com"
```

Действия

Действия отвечают за 'бизнес-логику' модуля. Действие обычно состоит из controller и action view.

Модуль может:

- Вызвать действия, которые уже определены в веб-интерфейсе Zabbix.
- Переопределить действия по умолчанию настраиваемыми действиями.
- Определить совершенно новые действия.

Чтобы переопределить поведение действия по умолчанию некоторым пользовательским поведением, определите действие с тем же именем в конфигурации модуля. При вызове действия вместо действия Zabbix по умолчанию будет выполнено действие модуля.

Файлы действий должны храниться в папке *actions*. Действия необходимо указать в файле manifest.json.

Контроллер

Рабочий процесс контроллера действия:

1. Проверьте, что все параметры, переданные в HTTP-запросе, являются допустимыми:

- Вызовите метод контроллера *checkInput()*;
- Используйте правила проверки, определенные в CNewValidator.php;
- Вызовите метод *validateInput()*.

2. Проверьте права пользователя.
3. Подготовьте данные в соответствии с переданными параметрами: если *checkInput()* возвращает true, Zabbix вызывает метод контроллера *doAction()*.
4. Подготовьте массив **$data** для представления. Используйте *CControllerResponseData* и метод *setResponse()* для сохранения ответа в массиве **$data**.

Пример:

```
/**
 * Validate input parameters.
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);
```

```
    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * Check user permissions.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Execute action and generate response object.
 */
protected function doAction(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
Полный список доступных классов контроллеров можно посмотреть в исходном коде Zabbix.

Представления

Файл представления получает данные от контроллера, а затем формирует HTML-отображение.

Note:
Определение представления(й) для модуля веб-интерфейса является необязательным, если только модуль не является виджетом.
 Виджеты панели управления должны иметь как минимум два представления: одно для режима редактирования и одно для режима просмотра (их следует хранить в каталоге *views*).

В представлении можно использовать предопределенные HTML-классы Zabbix (из */zabbix/ui/include/classes/html*), а также добавлять новые HTML- и CSS-классы. Новые классы следует хранить в папке *assets* модуля.

Пример:

```
...
(new CColHeader(_('Name')))
```

Это добавит новый столбец с именем *Name* и оформит верхнюю строку таблицы так же, как на других страницах Zabbix.

Представление действия

Это справочный файл для определения представления действий.

```
<?php declare(strict_types = 1);

/**
* @var CView $this
*/
```

```
$this->includeJsFile('example.something.view.js.php');

(new CWidget())
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
    'contacts' => $data['contacts']
    ])
    ->show();
```

Ресурсы

Каталог *assets* может содержать любые файлы и подкаталоги, которые не принадлежат другим директориям. Вы можете использовать её для:

- JavaScript стилей (должны находиться внутри *assets/js*);
- CSS стилей (должны находиться внутри *assets/css*);
- Изображений;
- Шрифтов;
- Всего, что вам нужно добавить.

assets/js

Каталог *assets/js* зарезервирован и должен содержать только JavaScript-файлы. Чтобы использовать их в виджете, укажите эти файлы в *manifest.json*.

Например:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css зарезервирован и должен содержать только файлы CSS стилей. Чтобы использовать их в виджете, укажите эти файлы в *manifest.json*.

Пример:

```
"assets": {
    "css": ["mywidget.css"]
}
```

Стили CSS

Файлы CSS могут содержать пользовательский атрибут `theme` для определения разных стилей для определённых тем веб-интерфейса.

Доступные темы и значения их атрибутов:

- **Синяя** - [theme='blue-theme']
- **Синяя (классическая)** - [theme='blue-classic-theme']
- **Тёмная** - [theme='dark-theme']
- **Тёмно-синяя** - [theme='dark-blue-theme']
- **Тёмная (классическая)** - [theme='dark-classic-theme']
- **С высокой контрастностью, светлая** - [theme='hc-light']
- **С высокой контрастностью, тёмная** - [theme='hc-dark']

Пример:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Зарегистрируем новый модуль

В этом разделе объясняется, как добавить новый модуль во внешний интерфейс Zabbix.

Предварительные требования Прежде чем продолжить, убедитесь, что:

- Модуль расположен внутри каталога *modules* вашей установки внешнего интерфейса Zabbix (например, *zabbix/ui/modules*).
- Модуль имеет как минимум базовую версию файла manifest.json.
- У вас есть доступ к разделу меню «Администрирование» в Zabbix (требуется тип роли пользователя «Супер-администратор»).

Note:
Интерфейс не будет устанавливать или даже распознавать несовместимые модули.

Добавление модуля Откройте страницу *Administration* > *General* > *Modules* и нажмите *Scan directory*.



Найдите свой модуль в списке и активируйте его.

Чтобы активировать модуль, нажмите на ссылку *Disabled* — состояние модуля изменится на *Enabled*.

Нажмите на имя модуля, чтобы просмотреть дополнительную информацию о модуле, такую как автор, версия или краткое описание (если они определены в манифесте).

Предварительный просмотр виджета Модули виджетов после добавления сразу становятся видны в списке виджетов на панели мониторинга.

Вы можете открыть панель мониторинга, переключиться в режим редактирования и добавить виджет на панель мониторинга обычным способом.

После внесения изменений в виджет обновите панель мониторинга, чтобы посмотреть, как виджет выглядит с учетом последних изменений.

Виджеты

Виджеты - это внешние модули Zabbix, используемые для информационных панелей. Если не указано иное, все рекомендации по модулям также применимы к виджетам.

Однако виджет заметно отличается от модуля. Чтобы создать виджет:

- указать тип ”виджет” в файле manifest.json (”тип”: ”виджет”);
- включить как минимум два представления: одно для режима представления виджета и одно для режима конфигурации виджета (example.widget.view.php и example.widget.edit.php);
- и контроллер для представления виджета (WidgetView.php);
- используйте и расширяйте стандартные классы виджетов.

Конфигурация

На этой странице описаны классы, которые можно использовать для создания представления конфигурации виджета с настраиваемыми полями конфигурации. Представление конфигурации виджета - это часть виджета, которая позволяет пользователю настраивать параметры виджета для презентации.

Виджет

Основной класс виджетов, расширяет базовый класс всех виджетов информационной панели - *CWidget*. Требуется для переопределения поведения виджета по умолчанию.

Класс *Widget* должен находиться в корневом каталоге виджета (например, *zabbix/ui/modules/my_custom_widget*).

Пример Widget.php

```php
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const MY_CONSTANT = 0;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

WidgetForm

Класс *WidgetForm* расширяет класс по умолчанию *CWidgetForm* и содержит набор полей *CWidgetField*, которые необходимы для определения структуры хранения конфигурации виджета в базе данных и обработки ввода.

Класс *WidgetForm* должен находиться в каталоге *includes*. Если класс имеет другое имя, это имя должно быть указано в параметре *widget/form_class* в файле *manifest.json*.

пример includes/WidgetForm.php

```php
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};

class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
                    ->setFilterParameter('numeric', true)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

Класс *CWidgetFormView* необходим для задания логики представления полей, определённых в классе *WidgetForm*, а также для определения их внешнего вида и поведения при отображении в представлении конфигурации.

Класс *CWidgetFormView* поддерживает следующие методы:

- *addField()* - принимает в качестве параметра экземпляр класса *CWidgetFieldView*; для каждого класса *CWidgetField* существует соответствующий класс *CWidgetFieldView*, используемый в представлении конфигурации виджета.
- *addFieldset()* - принимает экземпляр класса *CWidgetFieldsGroupView*, который объединяет поля в сворачиваемый контейнер.
- *addFieldsGroup()* - принимает экземпляр класса *CWidgetFormFieldsetCollapsibleView*, который визуально (с помощью границы) объединяет поля в группу.
- *includeJsFile()* - позволяет добавить файл JavaScript в представление конфигурации виджета.
- *addJavaScript()* - позволяет добавить встроенный JavaScript-код, который будет выполнен сразу после загрузки представления конфигурации виджета.

Класс *CWidgetFormView* должен находиться в каталоге *views*.

Пример views/widget.edit.php

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

Класс JavaScript можно использовать для добавления динамического поведения и интерактивности в представление конфигурации виджета. Например, вы можете инициализировать палитру цветов, определенную в классе *CWidgetFormView*.

Класс JavaScript должен быть загружен вместе с формой, поэтому на него следует ссылаться в классе *CWidgetFormView* с помощью методов *includeJsFile()* и *addJavaScript()*.

В приведенном ниже примере экземпляр одноэлементного класса сразу создается и сохраняется под именем *window.my_custom_widget_form*. Таким образом, открытие формы во второй раз приведет к воссозданию экземпляра.

Класс JavaScript должен находиться в каталоге *views*.

пример views/widget.edit.js.php

```
<?php
```

```
use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._advanced_configuration = document.getElementById('adv_conf');

        this._advanced_configuration.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
    }

    updateForm() {
        const show_advanced_configuration = this._advanced_configuration.checked;

        for (const element of this._form.querySelectorAll('.js-advanced-configuration')) {
            element.style.display = show_advanced_configuration ? '' : 'none';
        }
    }
};
```

CWidgetField

Класс *CWidgetField* является базовым классом, от которого наследуются все классы полей формы (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList* и т. д.). Классы, расширяющие *CWidgetField*, отвечают за получение, сохранение и проверку значений конфигурации виджета.

Доступны следующие классы *CWidgetField*.

Класс CWidgetField	Тип поля в базе данных	Описание
CWidgetFieldCheckBox	int32	Одиночный флажок.
CWidgetFieldCheckBoxList	array of int32	Несколько флажков в одном поле конфигурации.
CWidgetFieldColor	string	Поле выбора цвета.
CWidgetFieldDatePicker	string	Поле выбора даты.
CWidgetFieldIntegerBox	int32	Поле для ввода целого числа. Можно настроить минимальное и максимальное значения.
CWidgetFieldLatLng	string	Текстовое поле для ввода разделённых запятыми широты, долготы и уровня масштабирования карты.
CWidgetFieldMultiSelect	-	Базовый класс для полей множественного выбора, расширяемый всеми классами `CWidgetFieldMultiSelect*`.
CWidgetFieldMultiSelectActionId	ID	Поле множественного выбора действий из списка действий, определённых в разделе *Оповещения* > *Действия*.
CWidgetFieldMultiSelectGraphId	ID	Поле множественного выбора пользовательских графиков.
CWidgetFieldMultiSelectGraphPrototypeId	ID	Поле множественного выбора прототипов пользовательских графиков.
CWidgetFieldMultiSelectGroupId	ID	Поле множественного выбора групп узлов сети.
CWidgetFieldMultiSelectHostId	ID	Поле множественного выбора узлов сети.
CWidgetFieldMultiSelectHostInventory	array of int32	Поле множественного выбора полей инвентаризации узлов сети.

Класс CWidgetField	Тип поля в базе данных	Описание
CWidgetFieldMultiSelectItem	ID	Поле множественного выбора элементов данных.
CWidgetFieldMultiSelectItemPrototype	ID	Поле множественного выбора прототипов элементов данных.
CWidgetFieldMultiSelectMap	ID	Поле множественного выбора карт.
CWidgetFieldMultiSelectMediaType	ID	Поле множественного выбора типов носителей.
CWidgetFieldMultiSelectOverrideHost	ID	Поле множественного выбора источника данных для узлов сети из панели мониторинга или другого виджета.
CWidgetFieldMultiSelectService	ID	Поле множественного выбора сервисов.
CWidgetFieldMultiSelectSla	ID	Поле множественного выбора SLA.
CWidgetFieldMultiSelectUser	ID	Поле множественного выбора пользователей.
CWidgetFieldNavTree	array of (multiple mixed)	Хранит структуру дерева навигации карт для использования в виджете *Дерево навигации карт*.
CWidgetFieldNumericBox	string	Поле для ввода числа с плавающей запятой.
CWidgetFieldPatternSelect	-	Базовый класс для полей выбора по шаблону, расширяемый всеми классами `CWidgetPatternSelect*`.
CWidgetFieldPatternSelectHost	string	Поле множественного выбора узлов сети. Поддерживает шаблоны имён узлов сети.
CWidgetFieldPatternSelectItem	string	Поле множественного выбора элементов данных. Поддерживает шаблоны имён элементов данных.
CWidgetFieldRadioButtonList	int32	Группа переключателей, состоящая из одного или нескольких переключателей.
CWidgetFieldRangeControl	int32	Ползунок для выбора значения целочисленного типа.
CWidgetFieldReference	string	Хранит уникальный идентификатор этого виджета на панели мониторинга; используется для взаимодействия между виджетами.
CWidgetFieldSelect	int32	Раскрывающийся список.
CWidgetFieldSeverities	array of int32	Предварительно настроенный *CWidgetFieldCheckBoxList* с уровнями важности триггеров.
CWidgetFieldSparkline	array of (multiple mixed)	Позволяет настроить график-спарклайн.
CWidgetFieldTags	array of (string, int32, string)	Позволяет настроить одну или несколько строк фильтра тегов.
CWidgetFieldTextArea	string	Текстовая область для ввода многострочного текста.
CWidgetFieldTextBox	string	Текстовое поле для ввода однострочного текста.
CWidgetFieldThresholds	array of (string, string)	Позволяет настроить пары цветов и чисел.
CWidgetFieldTimePeriod	array of string	Поле выбора периода времени.
CWidgetFieldTimeZone	string	Раскрывающийся список с часовыми поясами.
CWidgetFieldUrl	string	Текстовое поле для ввода URL-адресов.

Note:
Максимальная длина полей `int32` составляет 10 символов, полей `ID` — 20 символов, а полей `string` — 65535 символов. Для составных типов (например, `array of (string, int32, string)`) ограничение применяется к каждому элементу отдельно. Ограничения длины для отдельных полей можно задать с помощью таких методов, как `setMaxLength()`.

Презентация

На этой странице описаны компоненты, которые можно использовать для создания представления виджета. Представление виджета - это часть виджета, которая получает данные в соответствии со своей конфигурацией и отображает их на информационной панели в контейнере.

Представление презентации состоит из трех частей:

- Действие виджета
- Просмотр виджета
- JavaScript

Действие виджета

Класс действий виджета (*WidgetView*) содержит методы для работы с виджетами в режиме просмотра презентации. Большинство действий виджетов используют и/или расширяют класс контроллера по умолчанию *CControllerDashboard-WidgetView*.

Класс действия виджета должен находиться в каталоге *actions* и указываться в параметре *actions* (*actions/widget.{id}.view/class*) в файле *manifest.json*.

Пример actions/WidgetView.php (реализован в родном для Zabbix виджете *Информация о системе*)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Виджет

Класс представления виджета (*CWidgetView*) отвечает за построение представления виджета.

Класс представления виджета должен находиться в каталоге *views*. Если файл, содержащий класс представления виджета, имеет имя, отличное от имени по умолчанию (*widget.view.php*), то его необходимо указать в файле *manifest.json* *actions* параметр (*actions/widget.{id}.view/view*).

пример views/widget.view.php

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

Класс JavaScript отвечает за определение поведения виджета, например, обновление данных виджета, изменение размера виджета, отображение элементов виджета и т. д.

Все операции JavaScript используют и/или расширяют базовый класс JavaScript всех виджетов панели - *CWidget*. Класс *CWidget* содержит набор методов с реализацией по умолчанию для поведения виджета. В зависимости от сложности виджета эти методы можно использовать как есть или расширить.

Класс *CWidget* содержит следующие методы:

- Методы, определяющие жизненный цикл виджета: *onInitialize()*, *onStart()*, *onActivate()*, *onDeactivate()*, *onDestroy()*, *onEdit()*.
- Методы, обрабатывающие обновление и отображение данных виджета: *promiseUpdate()*, *getUpdateRequestData()*, *processUpdateResponse(response)*, *processUpdateErrorResponse(error)*, *setContents(response)*.
- Методы, изменяющие внешний вид виджета: *onResize()*, *hasPadding()*.

Класс JavaScript должен находиться в каталоге *assets/js* и указываться в параметре *assets* (*assets/js*) в *manifest.json* файл.

Методы жизненного цикла

Методы жизненного цикла виджета вызываются панелью и на разных этапах жизненного цикла виджета во время его существования на панели.

Метод ***onInitialize()*** определяет исходное состояние и/или значения виджета без выполнения каких-либо манипуляций с HTML или данными. Этот метод вызывается при создании виджета (создается экземпляр объекта виджета), обычно

путем добавления виджета на страницу панели или загрузки страницы панели.

Пример:

```
onInitialize() {
    this._time_offset = 0;
    this._interval_id = null;
    this._clock_type = CWidgetClock.TYPE_ANALOG;
    this._time_zone = null;
    this._show_seconds = true;
    this._time_format = 0;
    this._tzone_format = 0;
    this._show = [];
    this._has_contents = false;
    this._is_enabled = true;
}
```

Метод ***onStart()*** определяет HTML-структуру виджета, не выполняя никаких манипуляций с данными. Этот метод вызывается перед первой активацией страницы панели мониторинга, то есть до того, как панель мониторинга и ее виджеты полностью отобразятся пользователю.

Пример:

```
onStart() {
    this._events.resize = () => {
        const padding = 25;
        const header_height = this._view_mode === ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER
            ? 0
            : this._header.offsetHeight;

        this._target.style.setProperty(
            '--content-height',
            `${this._cell_height * this._pos.height - padding * 2 - header_height}px`
        );
    }
}
```

Метод ***onActivate()*** делает виджет активным и интерактивным, включая настраиваемые прослушиватели событий (для реагирования на действия пользователя) и запуская цикл обновления виджета (для поддержания актуальности его содержимого). Этот метод вызывается при активации страницы информационной панели, то есть когда она полностью отображается в пользовательском интерфейсе.

Обратите внимание, что до вызова метода *onActivate()* виджет находится в неактивном состоянии («WIDGET_STATE_INACTIVE»). После успешного вызова виджет переходит в активное состояние («WIDGET_STATE_ACTIVE»). В активном состоянии виджет реагирует, слушает события, периодически обновляет свое содержимое и может взаимодействовать с другими виджетами.

Пример:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

Метод ***onDeactivate()*** останавливает любую активность и интерактивность виджета, деактивируя пользовательские прослушиватели событий и останавливая цикл обновления виджета. Этот метод вызывается, когда страница панели деактивируется, то есть отключается или удаляется, или когда виджет удаляется со страницы панели.

Обратите внимание, что до вызова метода *onDeactivate()* виджет находится в активном состоянии (WIDGET_STATE_ACTIVE). После успешного вызова виджет переходит в неактивное состояние («WIDGET_STATE_INACTIVE»).

Пример:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

Метод ***onDestroy()*** выполняет задачи очистки перед удалением виджета с панели управления, что может включать в себя закрытие соединения с базой данных, которое было установлено во время инициализации виджета, очистка временных данных для освобождения системной памяти и предотвращения утечек ресурсов, отмена регистрации прослушивателей событий, связанных с событиями изменения размера или нажатиями кнопок, чтобы предотвратить ненужную обработку событий и утечки памяти и т. д. Этот метод вызывается при удалении виджета или страницы информационной панели, которая его содержит.

Обратите внимание, что перед вызовом метода *onDestroy()* виджет в активном состоянии (`WIDGET_STATE_ACTIVE`) всегда деактивируется с помощью вызова метода *onDeactivate()*.

Пример:

```
onDestroy() {
    if (this._filter_widget) {
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

Метод ***onEdit()*** определяет внешний вид и поведение виджета, когда панель переходят в режим редактирования. Этот метод вызывается, когда панель переходит в режим редактирования, обычно при взаимодействии пользователя с кнопкой *Edit* у виджета или кнопкой *Edit dashboard* на панели.

Пример:

```
onEdit() {
    this._deactivateGraph();
}
```

Методы обновления виджета

Методы процесса обновления виджета отвечают за получение обновленных данных с сервера Zabbix или любого другого источника данных и их отображение в виджете.

Метод ***promiseUpdate()*** инициирует процесс обновления данных путем получения данных, обычно с помощью веб-запросов или вызовов API. Этот метод вызывается при отображении страницы панели и периодически после этого, пока страница панели не переключится на другую страницу панели.

Ниже приведен пример реализации по умолчанию метода *promiseUpdate()*, используемого большинством родных виджетов Zabbix. В реализации по умолчанию метод *promiseUpdate()* соответствует общему шаблону получения данных с сервера. Он создает новый объект Curl с соответствующим URL-адресом и параметрами запроса, отправляет запрос POST с помощью метода *fetch()* с объектом данных, созданным методом *getUpdateRequestData()*, и обрабатывает ответ (или ответ об ошибке) с помощью *processUpdateResponse(response)* или *processUpdateErrorResponse(error)* соответственно. Эта реализация подходит для большинства виджетов, поскольку они обычно получают данные в формате JSON и обрабатывают их согласованным образом.

```
promiseUpdate() {
    const curl = new Curl('zabbix.php');

    curl.setArgument('action', `widget.${this._type}.view`);

    return fetch(curl.getUrl(), {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify(this.getUpdateRequestData()),
        signal: this._update_abort_controller.signal
    })
        .then((response) => response.json())
        .then((response) => {
            if ('error' in response) {
                this.processUpdateErrorResponse(response.error);

                return;
            }

            this.processUpdateResponse(response);
        });
    }
```

Метод ***getUpdateRequestData()*** подготавливает данные запроса сервера для обновления виджета, собирая различные свойства и соответствующие им значения (идентификаторы виджета, настройки фильтра, временные диапазоны и т. д.) из состояния и конфигурации виджета, и создание объекта данных, который представляет необходимую информацию, подлежащую отправке на сервер в запросе на обновление. Этот метод вызывается только как часть метода *promiseUpdate()* по умолчанию, то есть во время процесса обновления виджета.

Реализация по умолчанию:

```
getUpdateRequestData() {
    return {
        templateid: this._dashboard.templateid ?? undefined,
        dashboardid: this._dashboard.dashboardid ?? undefined,
        widgetid: this._widgetid ?? undefined,
        name: this._name !== '' ? this._name : undefined,
        fields: Object.keys(this._fields).length > 0 ? this._fields : undefined,
        view_mode: this._view_mode,
        edit_mode: this._is_edit_mode ? 1 : 0,
        dynamic_hostid: this._dashboard.templateid !== null || this.supportsDynamicHosts()
            ? (this._dynamic_hostid ?? undefined)
            : undefined,
        ...this._contents_size
    };
}
```

Метод ***processUpdateResponse(response)*** обрабатывает ответ, полученный от сервера после запроса на обновление, и, если процесс обновления прошел успешно и без ошибок, очищает данные виджета и отображает новое содержимое с помощью метода *setContents()*. Этот метод вызывается только как часть метода *promiseUpdate()* по умолчанию, то есть во время процесса обновления виджета.

Реализация по умолчанию:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

Метод ***processUpdateErrorResponse(error)*** обрабатывает ответ, полученный от сервера после запроса на обновление, если ответ является ошибкой, и отображает сообщения об ошибках. Этот метод вызывается только как часть метода *promiseUpdate()* по умолчанию, то есть во время процесса обновления виджета.

Реализация по умолчанию:

```
processUpdateErrorResponse(error) {
    this._updateMessages(error.messages, error.title);
}
```

Метод ***setContents(response)*** отображает содержимое виджета, если процесс обновления виджета прошел успешно и без ошибок, что может включать в себя манипулирование элементами DOM, обновление компонентов пользовательского интерфейса, применение стилей или форматирования и т. д. Этот метод вызывается только как часть метода *processUpdateResponse(response)* по умолчанию, то есть в процессе обработки ответа, полученного от сервера после запроса на обновление.

Реализация по умолчанию:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

Методы модификации представления

Методы модификации представления виджета отвечают за изменение внешнего вида виджета.

Метод ***onResize()*** отвечает за настройку визуальных элементов виджета в соответствии с новым размером виджета, которые могут включать перестановку элементов, настройку размеров элементов, усечение текста, реализацию

отложенной загрузки для улучшения реагирования при изменении размера и т. д. Этот метод вызывается при изменении размера виджета, например, когда пользователь вручную изменяет размер виджета или когда изменяется размер окна браузера.

Пример:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

Метод ***hasPadding()*** отвечает за применение вертикального заполнения шириной 8 пикселей в нижней части виджета, когда он настроен на показ заголовка. Этот метод вызывается, когда страница информационной панели активируется, то есть когда она становится отображаемой страницей в пользовательском интерфейсе.

Реализация по умолчанию:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Для некоторых виджетов необходимо использовать все доступное пространство виджетов, например, для настройки собственного цвета фона. Ниже приведен пример реализации метода *hasPadding()*, используемого в родном для Zabbix виджете *значение элемента данных*.

```
hasPadding() {
    return false;
}
```

Руководства

Этот раздел содержит практические пошаговые руководства, иллюстрирующие, как создать собственный модуль и виджет в Zabbix.

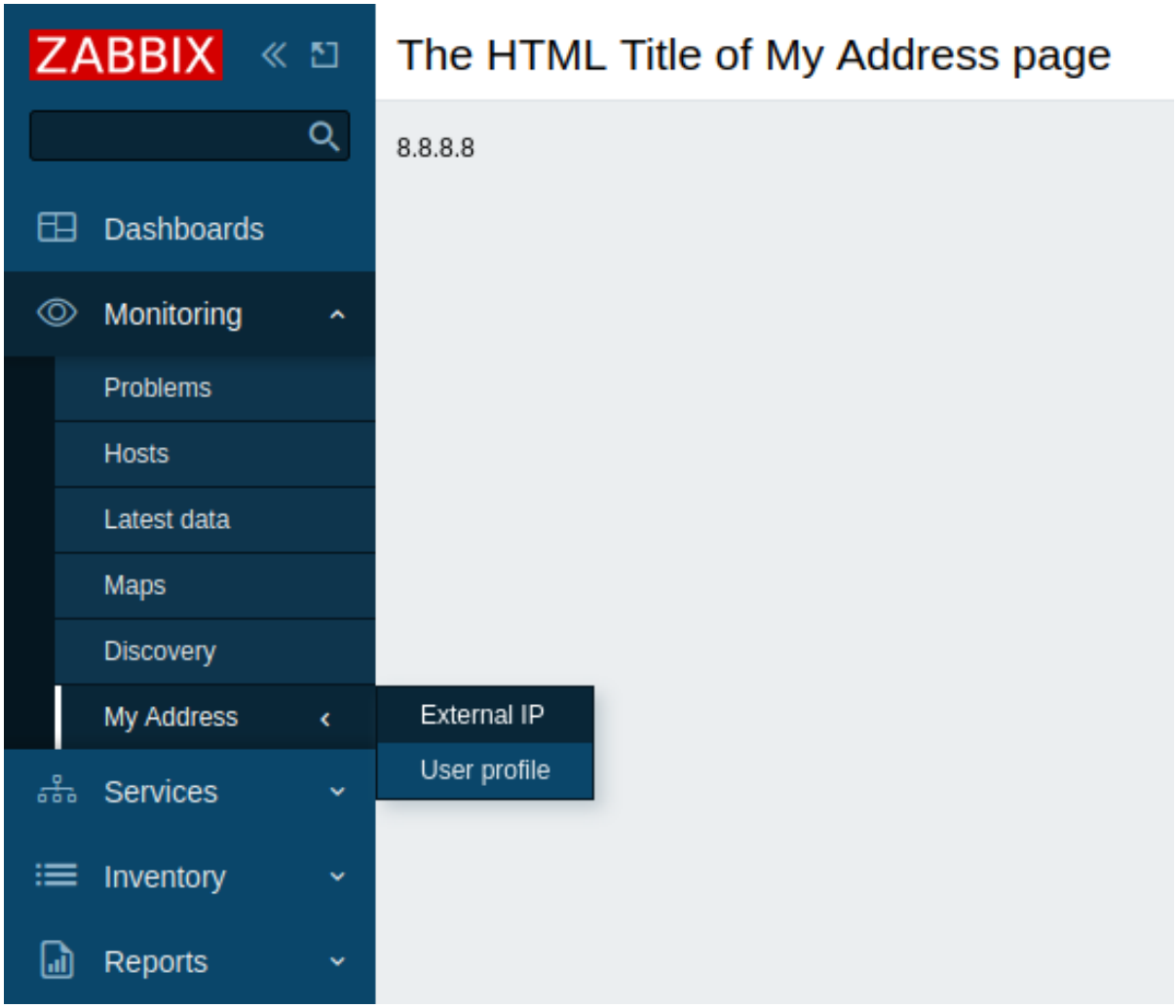
Создание модуля (руководство)

Эта пошаговая инструкция показывает, как создать простой модуль веб-интерфейса Zabbix. Вы можете скачать все файлы этого модуля в виде ZIP-архива: MyAddress.zip.

Что вы создадите

В ходе этого руководства вы сначала создадите модуль веб-интерфейса, который добавляет новый раздел меню *My Address*, а затем преобразуете его в более продвинутый модуль веб-интерфейса, который выполняет HTTP-запрос к *https://api.seeip.org* и отображает ответ - IP-адрес вашего компьютера - на новой странице в недавно созданном разделе меню *My Address*.

Вот как будет выглядеть готовый модуль:



Часть I - Новый раздел меню

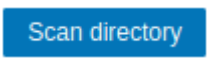
Добавление пустого модуля в веб-интерфейс Zabbix

1. Создайте каталог *MyAddress* в каталоге *modules* вашей установки веб-интерфейса Zabbix (например, *zabbix/ui/modules*).
2. Создайте файл *manifest.json* с базовыми метаданными модуля (см. описание поддерживаемых параметров).

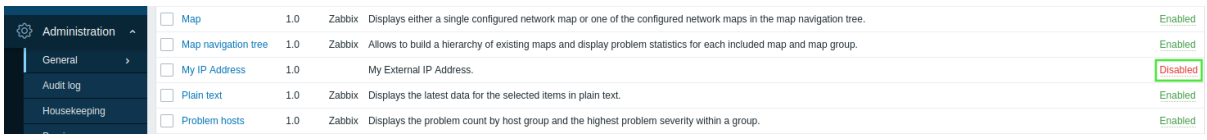
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. В веб-интерфейсе Zabbix перейдите в раздел *Administration* > *General* > *Modules* и нажмите кнопку *Scan directory*.



4. Найдите новый модуль *My IP Address* в списке и нажмите гиперссылку "Disabled", чтобы изменить статус модуля с "Disabled" на "Enabled" (если модуль не отображается в списке, см. раздел troubleshooting).



Теперь модуль зарегистрирован в веб-интерфейсе. Однако он пока не отображается, поскольку вам еще нужно определить его функциональность. После того как вы добавите содержимое в каталог модуля, изменения сразу станут

видны в веб-интерфейсе Zabbix после обновления страницы.

Создание раздела меню

1. Создайте файл *Module.php* в каталоге *MyAddress*.

Этот файл реализует новый класс *Module*, который наследуется от стандартного класса *CModule*. Класс *Module* добавит новый раздел меню *My Address* в главное меню.

Метод *setAction()* задает действие, которое будет выполнено при нажатии на раздел меню. Для начала можно использовать предопределенное действие *userprofile.edit*, которое откроет страницу *Профиль пользователя*. В части III этого руководства вы узнаете, как создать собственное действие.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

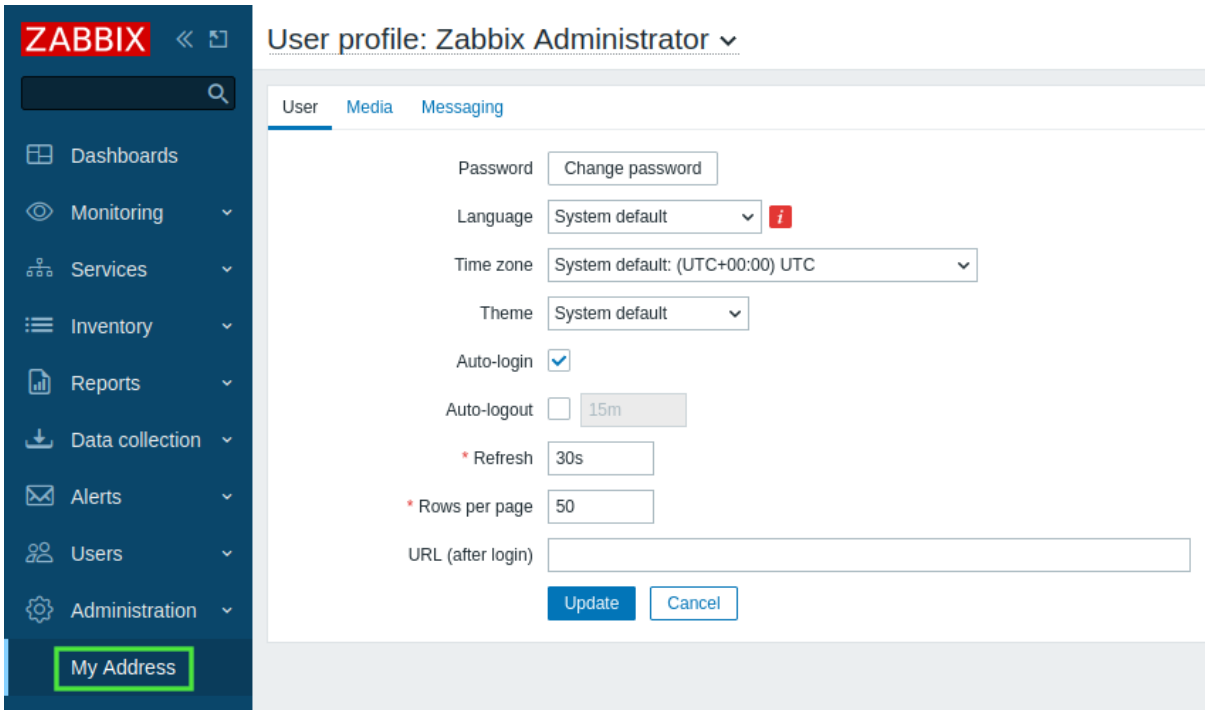
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Вы можете заменить `'userprofile.edit'` на другие действия, например, `'charts.view'` (открывает пользовательские графики), `'problems.view'` (открывает *Мониторинг* > *Проблемы*) или `'report.status'` (открывает отчет *Сведения о системе*).

3. Обновите веб-интерфейс Zabbix. Теперь в нижней части главного меню Zabbix появился новый раздел *My Address*. Нажмите *My Address*, чтобы открыть страницу *Профиль пользователя*.



Часть II - Изменение расположения раздела меню

В этой части вы переместите раздел меню *My Address* в раздел *Monitoring*, а затем добавите к нему вложенное меню. В результате пользователи смогут открывать две страницы подменю из раздела меню *Monitoring* > *My Address*.

1. Откройте и отредактируйте файл *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

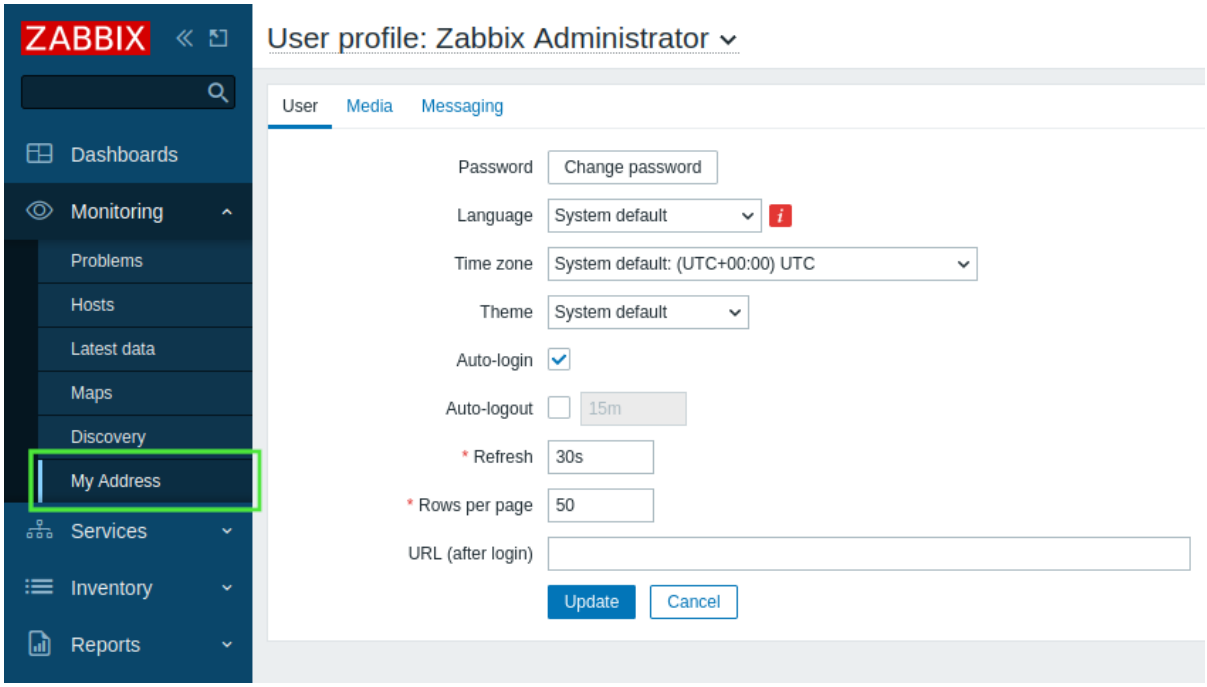
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Обновите веб-интерфейс Zabbix. Разверните раздел меню *Monitoring* и убедитесь, что раздел *My address* теперь расположен ниже раздела *Discovery*.



3. Чтобы добавить вложенные страницы в раздел меню *My Address*, снова откройте и отредактируйте файл *Module.php*.

Этот шаг создает два подраздела:

- *External IP*, который выполняет новое действие 'my.address', которое будет определено на следующих шагах;
- *User profile*, который выполняет предопределенное действие 'userprofile.edit' для открытия страницы *User profile*.

Обратите внимание, что для вложенного меню необходимо использовать класс *CMenu* в дополнение к классам, использованным на предыдущих шагах.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

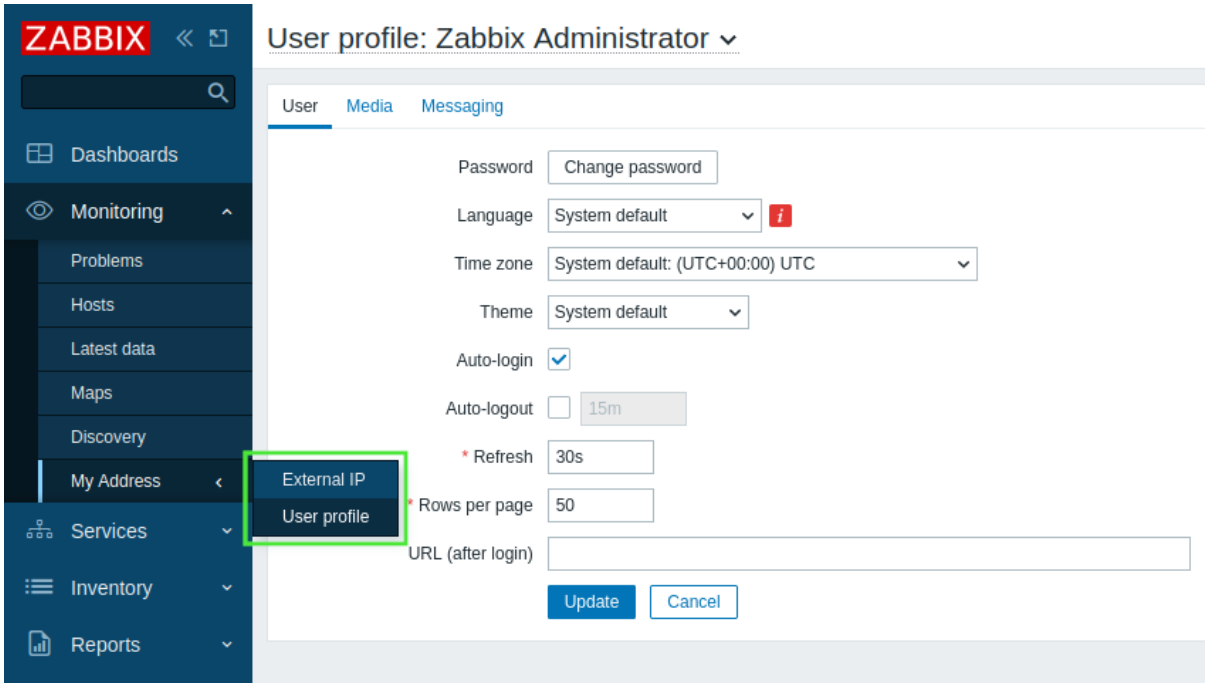
use Zabbix\Core\CModule,
    APP,
```

```
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Обновите веб-интерфейс Zabbix. Убедитесь, что раздел меню *My address* теперь содержит меню третьего уровня с двумя страницами - *External IP* и *User profile*.



Часть III - Действие модуля

Действие реализуется в двух файлах - *actions/MyAddress.php* и *views/my.address.php*. Файл ***actions/MyAddress.php*** отвечает за реализацию бизнес-логики, а файл ***views/my.address.php*** - за представление.

1. Создайте каталог *actions* в каталоге *MyAddress*.
2. Создайте файл *MyAddress.php* в каталоге *actions*.

Логика действия будет определена в классе *MyAddress*. Этот класс действия реализует четыре функции: *init()*, *checkInput()*, *checkPermissions()* и *doAction()*. Веб-интерфейс Zabbix вызывает функцию *doAction()* при запросе действия. Эта функция отвечает за бизнес-логику модуля.

Attention:
Данные должны быть организованы в виде ассоциативного массива. Массив может быть многомерным и содержать любые данные, ожидаемые представлением.

ui/modules/MyAddress/actions/MyAddress.php

```php
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```

3. Создайте новый каталог *views* в каталоге *MyAddress*.
4. Создайте файл *my.address.php* в каталоге *views* и определите представление модуля.

Обратите внимание, что переменная $data доступна в представлении без явного определения. Фреймворк автоматически передает ассоциативный массив в представление.

ui/modules/MyAddress/views/my.address.php

```php
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

5. Действие модуля должно быть зарегистрировано в файле *manifest.json*. Откройте *manifest.json* и добавьте новый объект `actions`, который содержит:

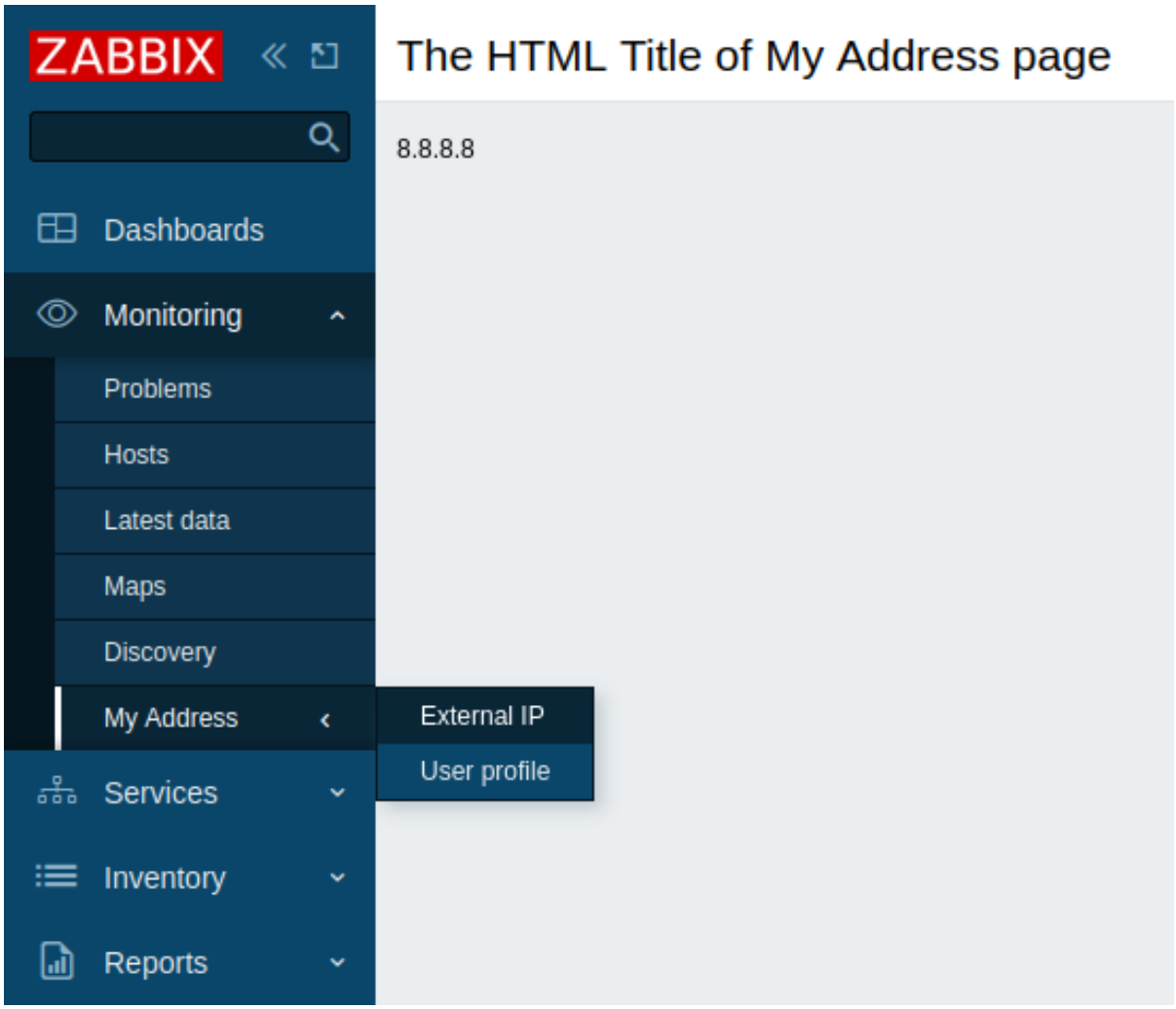
- ключ действия с именем действия, записанным в нижнем регистре (a-z) и с разделением слов точками (например, `my.address`);
- имя класса действия (`MyAddress`) в качестве значения ключа `class` объекта `my.address`;
- имя представления действия (`my.address`) в качестве значения ключа `view` объекта `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
```

```
    }
}
```

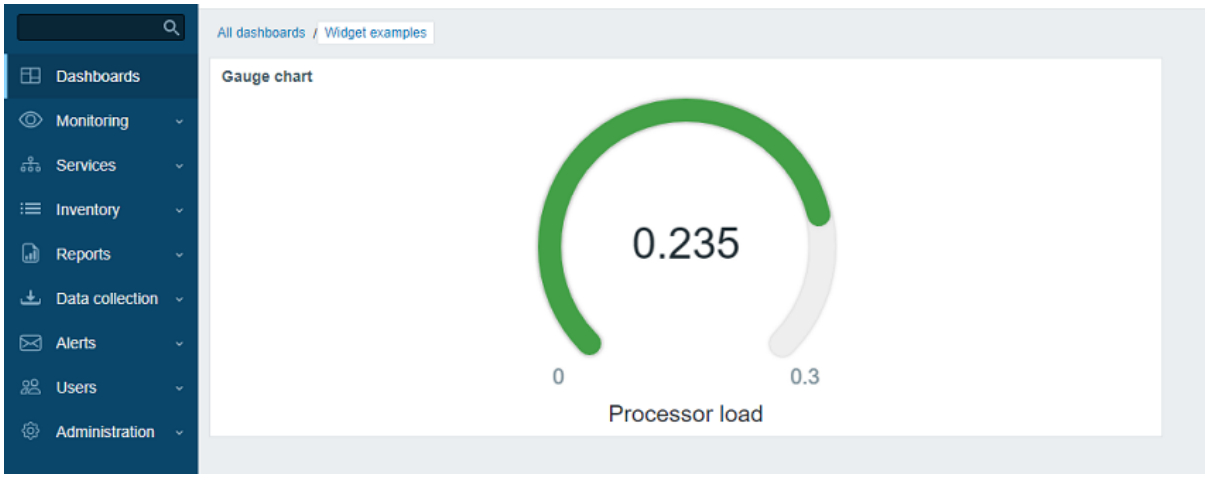
6. Обновите веб-интерфейс Zabbix. Нажмите *My address* > *External IP*, чтобы увидеть IP-адрес вашего компьютера.



Создание виджета (руководство)

Это пошаговое руководство, которое показывает, как создать простой виджет информационной панели. Вы можете скачать все файлы этого виджета в виде ZIP-архива: lesson_gauge_chart.zip.

Что вы создадите В ходе этого руководства вы сначала создадите базовый виджет "Hello, world!", а затем преобразуете его в более продвинутый виджет, который отображает значение элемента данных в виде диаграммы-датчика. Вот как будет выглядеть готовый виджет:



Часть I - ”Привет, мир!” В этом разделе вы изучите, как создать минимально необходимые элементы виджета и добавить новый виджет в интерфейс Zabbix.

Добавление пустого виджета во веб-интерфейс Zabbix

1. Создайте каталог `lesson_gauge_chart` в каталоге `modules` вашей установки веб-интерфейса Zabbix (например, *zabbix/ui/modules*).

> **Note:**
> Все пользовательские виджеты рассматриваются как внешние модули и должны быть добавлены в каталог `modules` вашей установки веб-интерфейса Zabbix (например, *zabbix/ui/modules*). Каталог `zabbix/ui/widgets` зарезервирован для встроенных виджетов Zabbix и обновляется вместе с Zabbix UI.

2. Создайте файл `manifest.json` с базовыми метаданными виджета (см. описание поддерживаемых параметров).

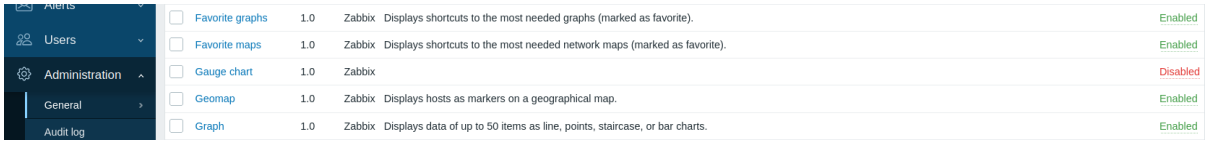
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

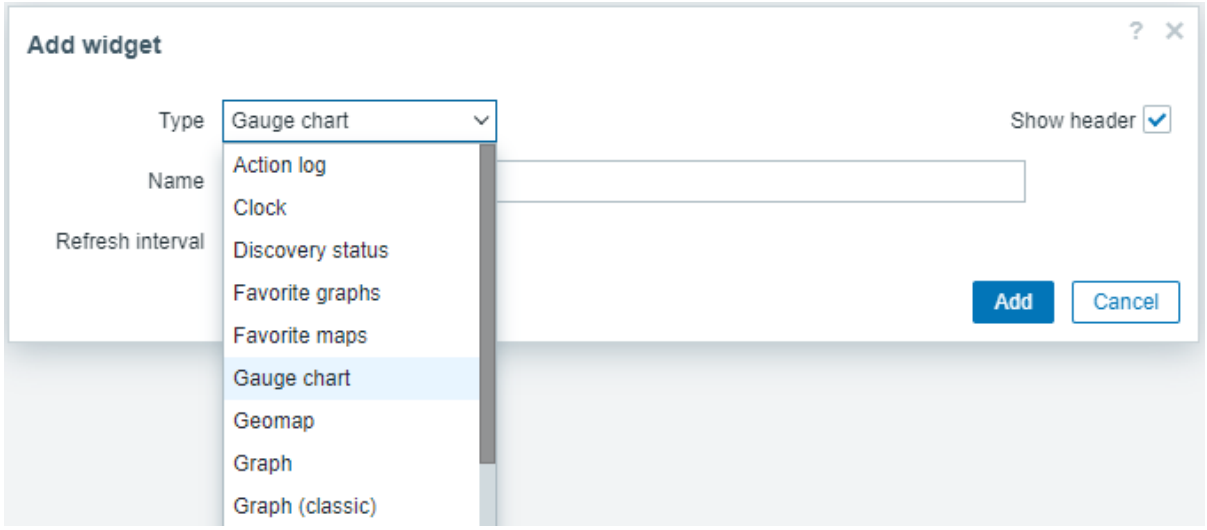
3. В веб-интерфейсе Zabbix перейдите в раздел *Administration* > *General* > *Modules* и нажмите кнопку *Scan directory*.



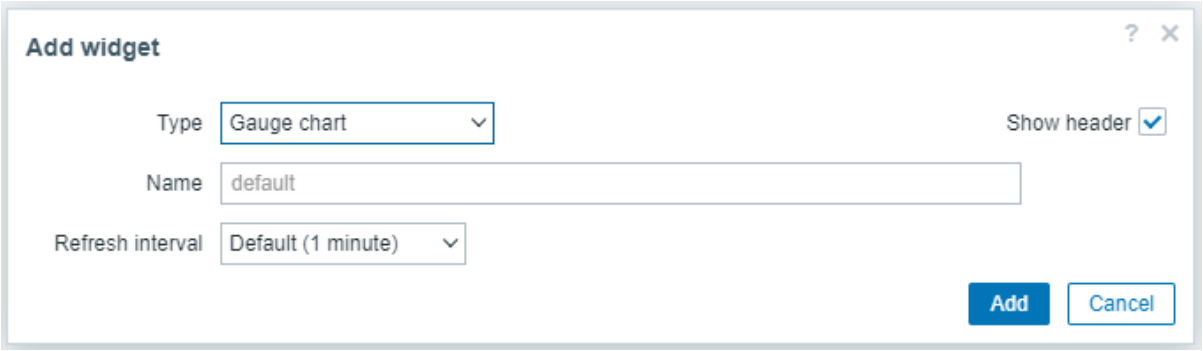
4. Найдите новый модуль *Gauge chart* в списке и нажмите гиперссылку ”Disabled”, чтобы изменить статус модуля с ”Disabled” на ”Enabled” (если модуль отсутствует в списке, см. раздел troubleshooting).



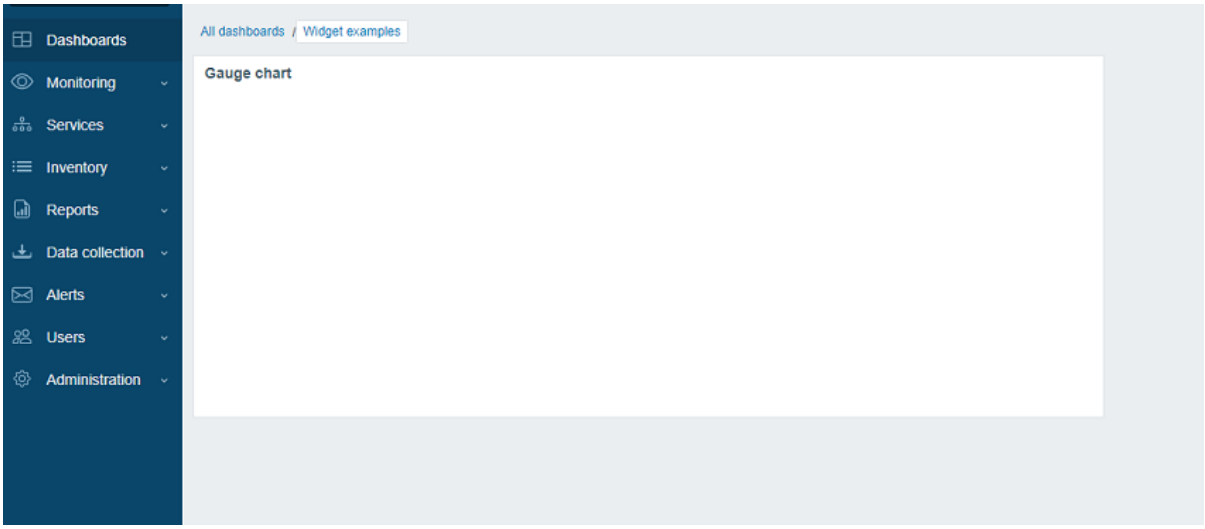
5. Откройте дашборд, переключите его в режим редактирования и добавьте новый виджет. В поле *Type* выберите ”Gauge chart”.



6. На этом этапе конфигурация виджета *Gauge chart* содержит только общие поля виджета *Name* и *Refresh interval*. Нажмите *Add*, чтобы добавить виджет на дашборд.



7. На дашборде должен появиться пустой виджет. Нажмите *Save changes* в правом верхнем углу, чтобы сохранить дашборд.



Добавление представления виджета

Note:
Файл **view** виджета должен находиться в каталоге `views` (для этого учебного примера: `ui/modules/lesson_gauge_chart/views/`). Если файл имеет имя по умолчанию `widget.view.php`, регистрировать его в файле `manifest.json` не нужно. Если файл имеет другое имя, укажите его в разделе `actions/widget.lesson_gauge_chart.view` файла `manifest.json`.

1. Создайте каталог `views` в каталоге `lesson_gauge_chart`.
2. Создайте файл `widget.view.php` в каталоге `views`.

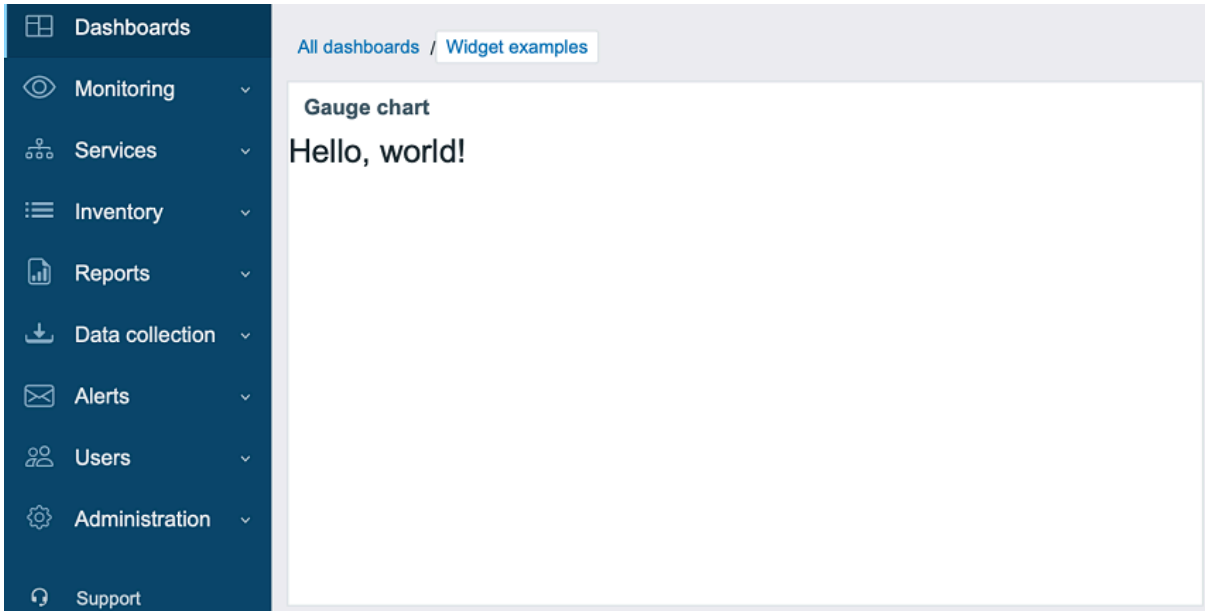
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 *              Gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Обновите панель мониторинга. Теперь виджет *Gauge chart* отображает "Hello, world!".



Часть II - Диаграмма измерений Добавление настроек в представление конфигурации и использование их в представлении виджета

В этом разделе вы узнаете, как добавить поле конфигурации виджета и вывести введенное значение в представлении виджета в виде текста.

Конфигурация виджета состоит из формы (`Zabbix\Widgets\CWidgetForm`) и представления формы виджета (`widget.edit.php`). Чтобы добавить поля (`Zabbix\Widgets\CWidgetField`), необходимо создать класс `WidgetForm`, который будет расширять `Zabbix\Widgets\CWidgetForm`.

Форма содержит набор полей (`Zabbix\Widgets\CWidgetField`) различных типов, которые используются для проверки значений, введенных пользователем. Поле формы (`Zabbix\Widgets\CWidgetField`) для каждого типа элемента ввода преобразует значение в единый формат для хранения в базе данных.

> **Note:**
> Файл **формы** виджета должен находиться в каталоге `includes` (для этого руководства: `ui/modules/lesson_gauge_chart/includes/`). Если файл имеет имя по умолчанию `WidgetForm.php`, регистрировать его в файле `manifest.json` не нужно. Если файл имеет другое имя, укажите его в разделе `widget/form_class` файла `manifest.json`.

1. Создайте новый каталог `includes` в каталоге `lesson_gauge_chart`.
2. Создайте файл `WidgetForm.php` в каталоге `includes`.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Добавьте поле *Description* в форму конфигурации виджета. Это обычное текстовое поле, в которое пользователь может вводить любые символы. Для этого можно использовать класс `CWidgetFieldTextBox`.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

use Zabbix\Widgets\Fields\CWidgetFieldTextBox;
```

```
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
               new CWidgetFieldTextBox('description', _('Description'))
            );
   }
}
```

4. В каталоге `views` создайте файл представления конфигурации виджета `widget.edit.php` и добавьте представление для нового поля *Description*. Для класса поля `CWidgetFieldTextBox` используется представление `CWidgetFieldTextBoxVie`

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

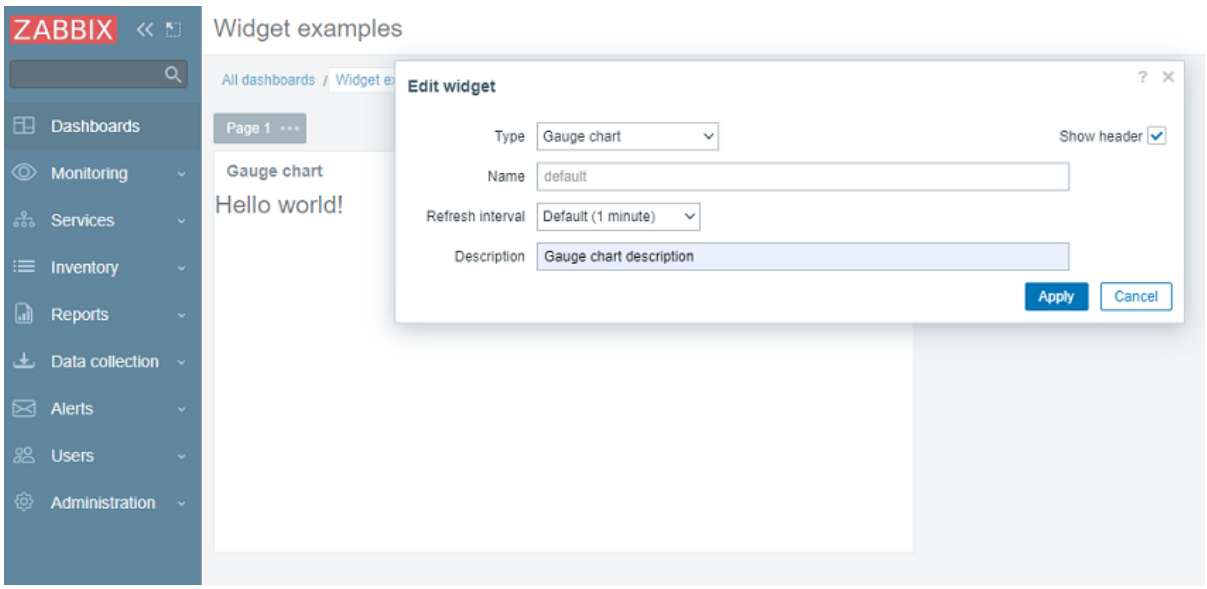
```
<?php

/**
 *                  Gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Перейдите на панель и нажмите значок шестеренки в виджете, чтобы открыть форму конфигурации виджета.

6. Теперь форма конфигурации виджета содержит новое текстовое поле *Description*. Введите любое значение, например *Gauge chart description*.



7. Нажмите *Apply* в форме конфигурации виджета. Затем нажмите *Save changes* в правом верхнем углу, чтобы сохранить панель. Обратите внимание, что новое описание нигде не отображается, и виджет по-прежнему показывает "Hello, world!".

Чтобы новое описание появилось в виджете, значение поля *Description* нужно получить из базы данных и передать в представление виджета. Для этого необходимо создать класс действия.

8. Создайте новый каталог `actions` в каталоге `lesson_gauge_chart`.

9. Создайте файл `WidgetView.php` в каталоге `actions`. Класс действия `WidgetView` будет расширять класс `CControllerDashboardWidgetView`.

Значения полей конфигурации виджета хранятся в свойстве `$fields_values` класса действия.

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. Откройте `manifest.json` и зарегистрируйте `WidgetView` как класс действия в разделе `actions/widget.lesson_gauge_cha`

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. Теперь вы можете использовать значение поля description, содержащееся в `$data['description']`, в представлении виджета. Откройте `views/widget.view.php` и замените статический текст "Hello, world!" на `$data['description']`.

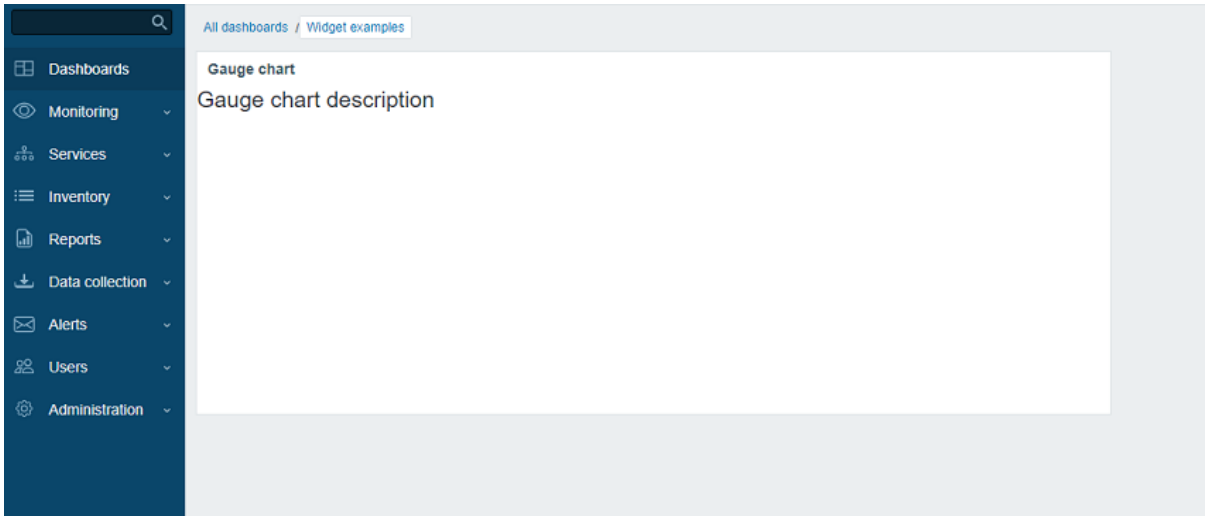
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 *              Gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Обновите страницу панели. Теперь вместо "Hello, world!" вы должны увидеть текст описания виджета.



Получение значения элемента данных через API

Виджет должен отображать последнее значение элемента данных, выбранного пользователем. Для этого нужно добавить возможность выбора элементов данных в конфигурации виджета.

В этом разделе вы узнаете, как добавить поле выбора элемента данных в форму виджета и как добавить визуальную часть этого поля в представление конфигурации. Затем контроллер виджета сможет получать данные элемента и его значение через запрос API. После получения значение можно отобразить в представлении виджета.

1. Откройте `includes/WidgetForm.php` и добавьте поле `CWidgetFieldMultiSelectItem`. Это позволит выбирать элемент данных в форме конфигурации.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 *               -    .
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. Откройте `views/widget.edit.php` и добавьте визуальный компонент поля в представление конфигурации.

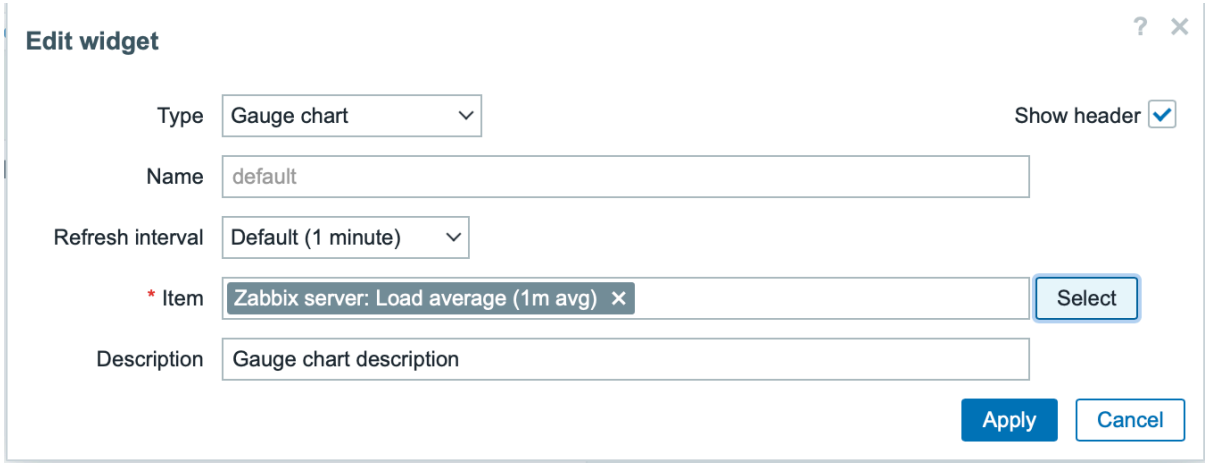
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```php
<?php

/**
 *                      -    .
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. Вернитесь на панель и нажмите значок шестеренки в виджете, чтобы открыть форму конфигурации виджета.
4. Теперь форма конфигурации виджета содержит новое поле ввода *Item*. Выберите узел сети "Zabbix server" и элемент данных "Load average (1m avg)".



5. Нажмите *Apply* в форме конфигурации виджета. Затем нажмите *Save changes* в правом верхнем углу, чтобы сохранить панель.
6. Откройте и измените `actions/WidgetView.php`.

Теперь идентификатор элемента данных будет доступен в контроллере виджета в `$this->fields_values['itemid']`. Метод контроллера `doAction()` получает данные элемента (имя, тип значения, единицы измерения) с помощью метода API `item.get` и последнее значение элемента с помощью метода API `history.get`.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
```

```
            ]
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. Откройте `views/widget.view.php` и добавьте значение элемента данных в представление виджета.

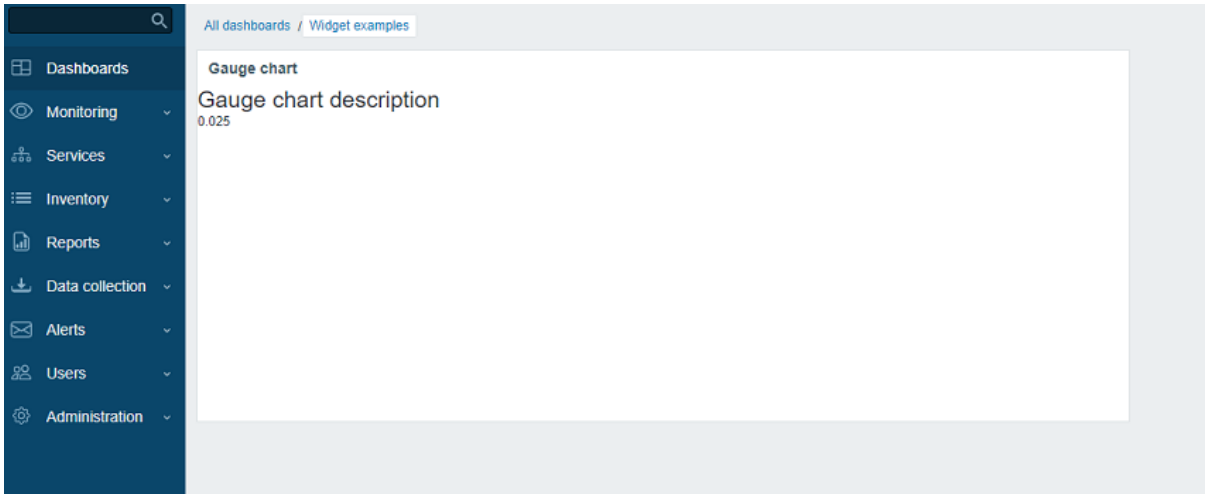
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 *                  -    .
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. Обновите страницу панели. Виджет будет отображать последнее значение элемента данных.



Добавление расширенных настроек конфигурации в представление конфигурации

В этом разделе вы узнаете, как добавить сворачиваемый/разворачиваемый раздел *Advanced configuration* с необязательными параметрами, такими как цвет, минимальное и максимальное значения, единицы измерения и поле *Description*, созданное ранее.

1. Создайте файл `Widget.php` в основном каталоге виджета `lesson_gauge_chart`, чтобы создать новый класс `Widget`.

Класс `Widget` будет расширять базовый класс `CWidget`, чтобы добавлять/переопределять настройки виджета по умолчанию (в данном случае - переводы). JavaScript, приведенный ниже, отображает строку "No data" в случае отсутствия данных. Строка "No data" присутствует в файлах перевода интерфейса Zabbix.

Если у виджета есть какие-либо константы, рекомендуется также указать их в классе `Widget`.

`ui/modules/lesson_gauge_chart/Widget.php`

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. Откройте `includes/WidgetForm.php` и добавьте новые поля *Color* (выбор цвета), *Min* (числовое поле), *Max* (числовое поле) и *Units* (список выбора), а также задайте палитру цветов по умолчанию для выбора цвета, чтобы ее можно было использовать на следующих шагах.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
```

```
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Откройте `views/widget.edit.php` и добавьте визуальные компоненты полей в представление конфигурации.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
```

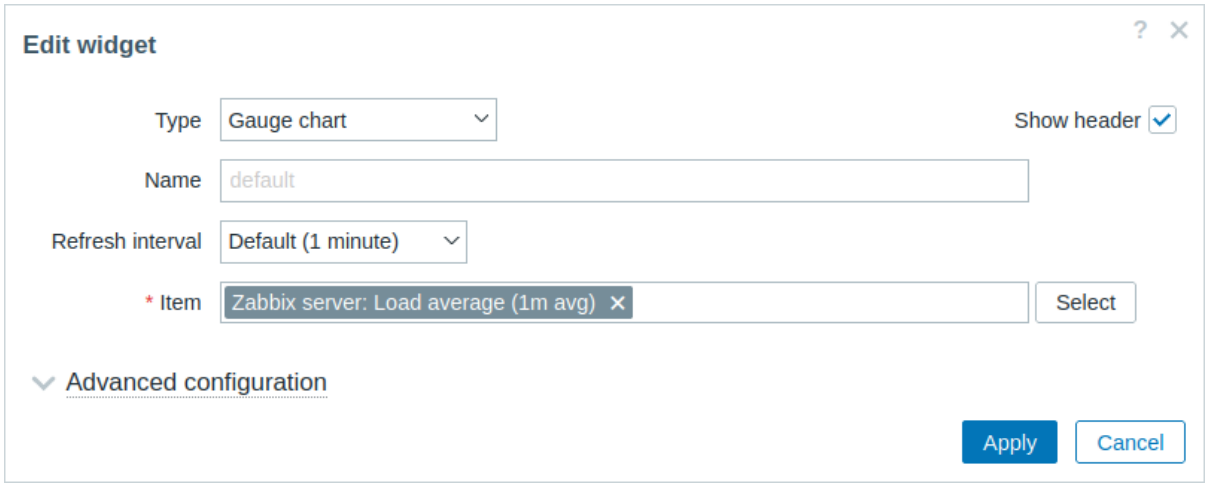
```
 *
 * @var CView $this
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

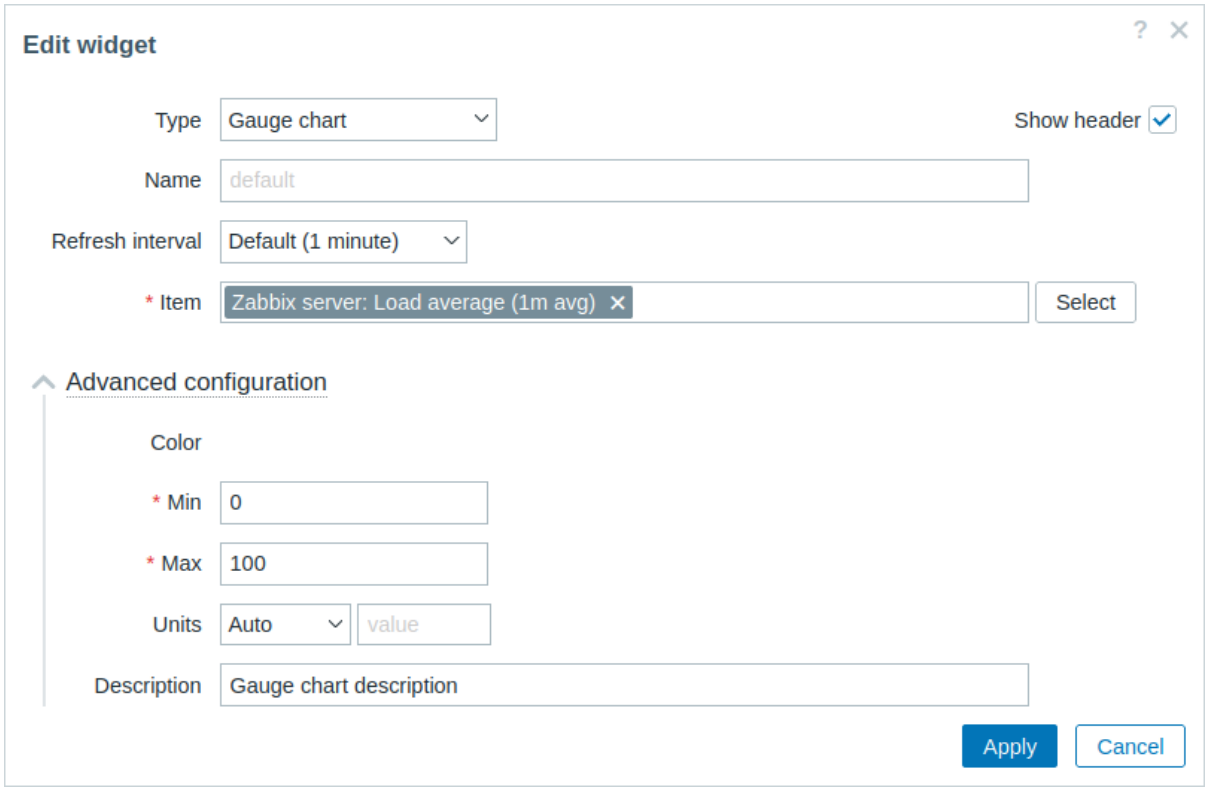
Note:
Метод `addField()` класса `CWidgetFormView` принимает строку с CSS-классом в качестве второго параметра.

4. Вернитесь на панель, переключитесь в режим редактирования и нажмите значок шестеренки в виджете, чтобы открыть форму конфигурации виджета. Теперь форма конфигурации виджета содержит новый сворачиваемый/разворачиваем раздел *Advanced configuration*.



5. Разверните раздел *Advanced configuration*, чтобы увидеть дополнительные поля конфигурации виджета. Обратите

внимание, что у поля *Color* пока нет выбора цвета. Это связано с тем, что выбор цвета должен инициализироваться с помощью JavaScript, который будет добавлен в следующем разделе - *Добавление JavaScript в виджет*.



Добавление JavaScript в виджет

В этом разделе вы узнаете, как добавить диаграмму-датчик, созданную с помощью JavaScript, которая показывает, является ли последнее значение нормальным или слишком высоким/слишком низким.

1. Создайте файл `widget.edit.js.php` в каталоге `views`.

JavaScript будет отвечать за инициализацию палитры цветов в окне настройки.

`ui/modules/lesson_gauge_chart/views/widget.edit.js.php`

```
<?php

use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._unit_select = document.getElementById('value_units');
        this._unit_value = document.getElementById('value_static_units');

        this._unit_select.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
```

```
    }

    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. Откройте views/widget.edit.php и добавьте файл widget.edit.js.php с JavaScript в окно настройки. Для этого используйте метод includeJsFile(). Чтобы добавить встроенный JavaScript, используйте метод addJavaScript().

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

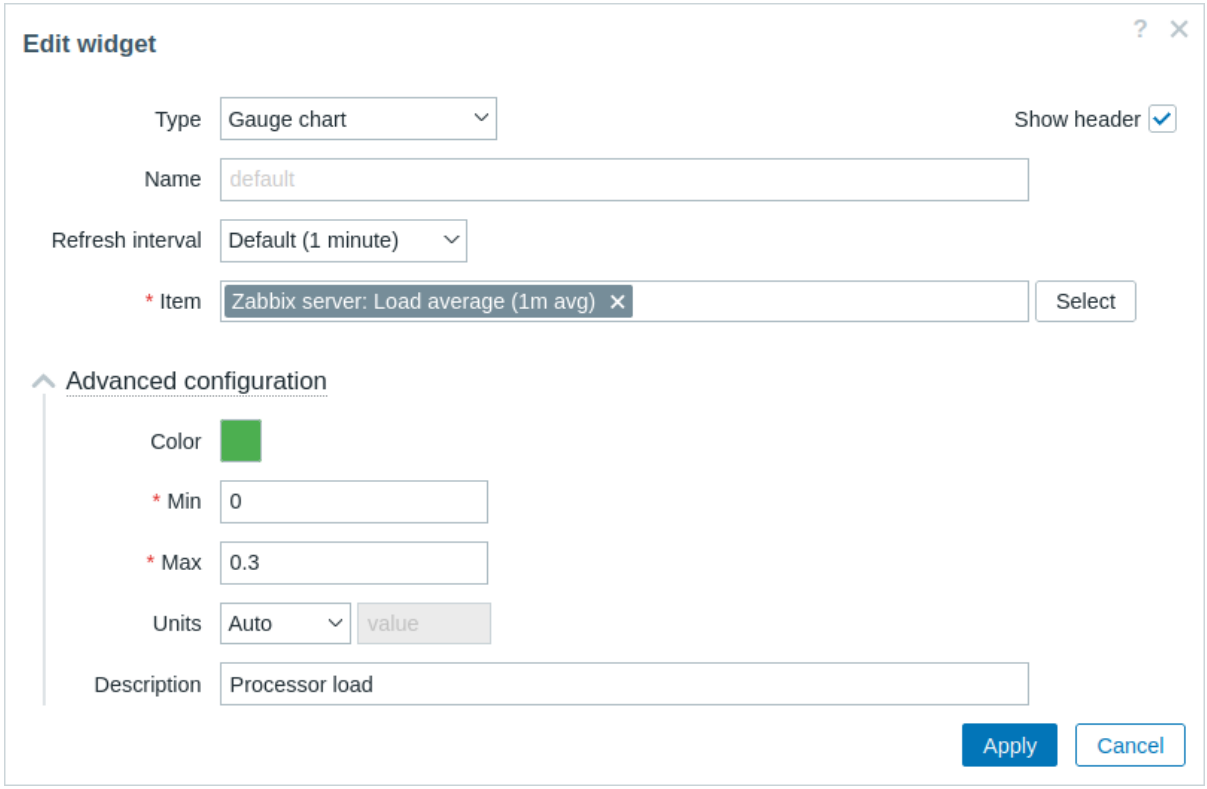
use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. Вернитесь на панель, нажмите значок шестеренки в виджете, чтобы открыть форму настройки виджета. Теперь разверните раздел *Advanced configuration*, чтобы увидеть инициализированную палитру цветов. Заполните поля

значениями и выберите цвет для диаграммы-датчика.



4. Нажмите *Apply* в форме настройки виджета. Затем нажмите *Save changes* в правом верхнем углу, чтобы сохранить панель.

5. Откройте `actions/WidgetView.php` и обновите контроллер.

Свойство `$this->fields_values` теперь содержит значения всех полей *Advanced configuration*. Завершите настройку контроллера, чтобы передавать конфигурацию и выбранное значение элемента данных в представление виджета.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
```

```
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. Откройте и измените `views/widget.view.php`.

Вам нужно создать контейнер для диаграммы-датчика, которую вы нарисуете на следующих шагах, и контейнер для описания.

Чтобы передать значения в JavaScript в виде объекта JSON, используйте метод `setVar()`.

`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. Создайте новый каталог `assets` в каталоге `lesson_gauge_chart`. Этот каталог будет использоваться для хранения JavaScript, CSS и, возможно, других ресурсов, таких как шрифты или изображения.
8. Для JavaScript представления виджета создайте каталог `js` в каталоге `assets`.
9. Создайте файл `class.widget.js` в каталоге `assets/js`.

Этот JavaScript-класс виджета будет расширять базовый JavaScript-класс всех виджетов панели - `CWidget`.

Панель зависит от корректной реализации виджета и передает всю необходимую информацию в виджет через вызов соответствующих JavaScript-методов. Панель также ожидает, что виджет будет генерировать события при возникновении определенного взаимодействия. Поэтому класс `CWidget` содержит набор методов с реализацией поведения виджета по умолчанию, которое можно настраивать путем расширения класса.

В данном случае требуется некоторая настройка, поэтому для следующего поведения виджета будет реализована пользовательская логика:

- инициализация виджета, отвечающая за определение начального состояния виджета (см. метод `onInitialize()`);
- отображение содержимого виджета (то есть отрисовка диаграммы-датчика), если процесс обновления виджета завершился успешно и без ошибок (см. метод `processUpdateResponse(response)` и связанные с ним методы `_resizeChart()` и `_updatedChart()`)
- изменение размера виджета (см. метод `onResize()` и связанный с ним метод `_resizeChart()`)

Для остальных аспектов виджета диаграммы-датчика будет использоваться реализация поведения виджета по умолчанию. Подробнее о JavaScript-методах класса `CWidget` см. в разделе: JavaScript.

Поскольку этот JavaScript требуется для представления виджета, он должен загружаться вместе со страницей панели. Чтобы включить загрузку JavaScript, необходимо обновить параметры `assets/js` и `js_class` в файле `manifest.json`, как показано в шаге 10.

ui/modules/lesson_gauge_chart/assets/js/class.widget.js

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');
```

```
            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
        }

        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;
```

```
                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);

                ctx.beginPath();
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. Откройте `manifest.json` и добавьте:

- имя файла (`class.widget.js`) в массив в разделе `assets/js`;
- имя класса (`WidgetLessonGaugeChart`) в параметр `js_class` в разделе `widget`.

Теперь класс `WidgetLessonGaugeChart` будет автоматически загружаться вместе с панелью.

`ui/modules/lesson_gauge_chart/manifest.json`

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

Добавление CSS-стилей к виджету

В этом разделе вы узнаете, как добавить пользовательские CSS-стили, чтобы сделать виджет более привлекательным.

1. Для стилей виджета создайте новый каталог css в каталоге assets.
2. Создайте файл widget.css в каталоге assets/css. Чтобы задать стили для элементов виджета, используйте селектор div.dashboard-widget-{widget id}. Чтобы настроить CSS для всего виджета, используйте селектор form.dashboard-widget-{widget id}

ui/modules/lesson_gauge_chart/assets/css/widget.css

```
div.dashboard-widget-lesson_gauge_chart {
    display: grid;
    grid-template-rows: 1fr;
    padding: 0;
}

div.dashboard-widget-lesson_gauge_chart .chart {
    display: grid;
    align-items: center;
    justify-items: center;
}

div.dashboard-widget-lesson_gauge_chart .chart canvas {
    background: white;
}

div.dashboard-widget-lesson_gauge_chart .description {
    padding-bottom: 8px;
    font-size: 1.750em;
    line-height: 1.2;
    text-align: center;
}

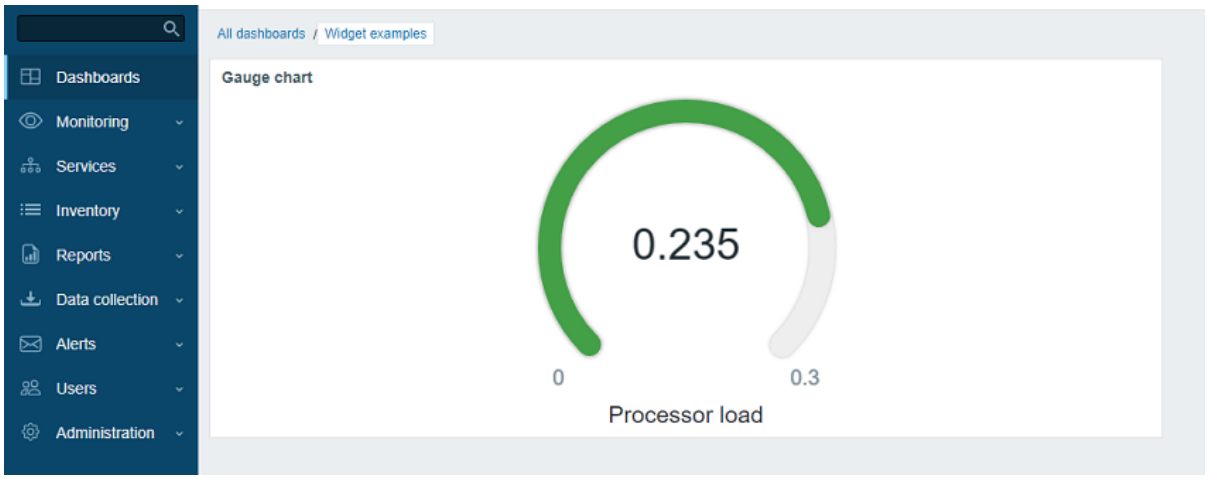
.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

3. Откройте manifest.json и добавьте имя файла CSS (widget.css) в массив в разделе assets/css. Это позволит CSS-стилям, определенным в widget.css, загружаться вместе со страницей панели мониторинга.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Обновите страницу панели мониторинга, чтобы увидеть готовую версию виджета.



Примеры

В этом разделе представлены файлы образцов модулей и виджетов, которые вы можете использовать в качестве основы для своих собственных модулей.

Чтобы использовать модуль:

1. Загрузите ZIP-архив.
2. Распакуйте содержимое в отдельную папку внутри папки /zabbix/ui/modules.
3. Зарегистрируйте модуль в веб-интерфейсе Zabbix.

Пример модуля

- При создании группы хостов предоставьте разрешения на чтение настроенным группам пользователей - hg_auto_perm.zip

Примеры виджетов

- Минимальный виджет - widget_min.zip
- Виджет «Привет, мир» с использованием только CSS — hello_world_css.zip
- Виджет «Привет, мир» с использованием только JavaScript — hello_world_js.zip
- Виджет «Привет, мир» с использованием PHP — hello_world_php.zip

Вы также можете использовать родные виджеты Zabbix в качестве примеров.

Изменения в разработке расширений

Изменения с 7.4 до 8.0

Страницы помощи Zabbix

Справочная документация (manpages) по инструментам и службам командной строки Zabbix, содержащая список доступных команд, параметры и примеры использования.

zabbix_agent2

```
ZABBIX_AGENT2(8)                                          ZABBIX_AGENT2(8)


       zabbix_agent2 — Zabbix     2


       zabbix_agent2 [-c     -  ]
       zabbix_agent2 [-c     -  ] [-v] -p
       zabbix_agent2 [-c     -  ] [-v] -t    -     -
       zabbix_agent2 [-c     -  ] -R runtime-
       zabbix_agent2 [-c     -  ] -T
       zabbix_agent2 -h
       zabbix_agent2 -V


       zabbix_agent2 —                                    .


       -c, --config     -
                                                          .

       -R, --runtime-control runtime-
                                          runtime-    .


              userparameter_reload


              log_level_increase


              log_level_decrease


              help


              metrics


              version


       -p, --print
                                       .
                                         ,
```

.

.

,

. t,

. ,

, ,

Zabbix zabbix_get,

. :

d .

m . ,

( )

, .

.

s . .

t . , s.

u .

-t, --test - -

. --print .

-v, --verbose

.

-T, --test-config

.

-h, --help

.

-V, --version

.

/usr/local/etc/zabbix_agent2.conf

Zabbix 2 (

).

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_AGENT2(8)

zabbix_agentd

ZABBIX_AGENTD(8) ZABBIX_AGENTD(8)

zabbix_agentd — Zabbix

```
zabbix_agentd [-c    -  ]
zabbix_agentd [-c    -  ] -p
zabbix_agentd [-c    -  ] -t   -    -
zabbix_agentd [-c    -  ] -R runtime-
zabbix_agentd [-c    -  ] -T
zabbix_agentd -h
zabbix_agentd -V
```

zabbix_agentd — .

-c, --config -
.

-f, --foreground
Zabbix .

-R, --runtime-control runtime-
runtime- .

userparameter_reload

log_level_increase[=]
, ,

log_level_decrease[=]
, ,

-
(active checks, collector,
listener)

- ,N
(: listener,3)

pid , 65535. 65535
« - ,N»

-p, --print
.
,
.

.
,
. t,
. ,
, ,
Zabbix zabbix_get,
. :

d .

m . ,
( )
, .

.

s . .

t . , s.

u .

-t, --test - -
. --print .

-T, --test-config
.

-h, --help
.

-V, --version
.

/usr/local/etc/zabbix_agentd.conf
Zabbix (
).

zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8),
zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_AGENTD(8)

zabbix_get

ZABBIX_GET(1) ZABBIX_GET(1)

zabbix_get — Zabbix get

zabbix_get -s - - -IP [-p -] [-I IP-] [-t
-] -k - -
zabbix_get -s - - -IP [-p -] [-I IP-] [-t
-] --tls-connect cert --tls-ca-file CA- [--tls-crl-file
CRL-] [--tls-agent-cert-issuer] [--tls-agent-cert-subject
] --tls-cert-file cert- --tls-key-file key-
[--tls-cipher13 -] [--tls-cipher -] -k - -
zabbix_get -s - - -IP [-p -] [-I IP-] [-t
-] --tls-connect psk --tls-psk-identity -PSK
--tls-psk-file PSK- [--tls-cipher13 -] [--tls-cipher
-] -k - -
zabbix_get -h
zabbix_get -V

zabbix_get — Zabbix .

-s, --host - - -IP
IP .

-p, --port -
, . :
10050.

-I, --source-address IP-
IP .

-t, --timeout
. : 1-600 (: 30)

-k, --key - -
, .

-P, --protocol
, . :

auto JSON,
,
( )

json JSON

plaintext ,
(6.4.x)

--tls-connect
. :

unencrypted
( )

psk TLS (pre-shared key, PSK)

cert TLS

--tls-ca-file CA-
, CA
.

--tls-crl-file CRL-
, .

--tls-agent-cert-issuer
.

--tls-agent-cert-subject
.

--tls-cert-file cert-
,
.

--tls-key-file key-
, (private key).

--tls-psk-identity -PSK
, (PSK).

--tls-psk-file PSK-
, (pre-shared key).

--tls-cipher13 -

OpenSSL 1.1.1 TLS 1.3.
. ,
OpenSSL 1.1.1.

--tls-cipher -
GnuTLS (TLS 1.2)
OpenSSL (TLS 1.2).
.

-h, --help
.

-V, --version
.

0, zabbix_get
Zabbix . , 1.

zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
--tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file
--tls-agent-cert-issuer "CN=Signing CA,OU=IT operations,O=Example
Corp,DC=example,DC=com" --tls-agent-cert-subject "CN=server1,OU=IT op-
erations,O=Example Corp,DC=example,DC=com" --tls-cert-file /home/zab-
bix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
--tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd"
--tls-psk-file /home/zabbix/zabbix_agentd.psk

zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8),
zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_GET(1)

zabbix_js

ZABBIX_JS(1) ZABBIX_JS(1)

zabbix_js — Zabbix JS

zabbix_js -s - -p - [-l -] [-t -]
zabbix_js -s - -i - [-l -] [-t -]
zabbix_js -h
zabbix_js -V

zabbix_js — ,
.

-s, --script -
.
«-», stdin.

```
       -p, --param     -
                            ,            .
                                                  «value».
                                  «-i».

       -i, --input     -
                     ,                ,            .

                        «value».
              «-»,               stdin.                    «-p».

       -w, --webdriver url
                   URL   -     .

       -l, --loglevel     -
                               . (     : 0-5).

       -t, --timeout    -
                               .           : 1-600
              (       : 10)

       -h, --help
                                        .

       -V, --version
                                    .


       zabbix_js -s script-file.js -p example




       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                              2022-01-06                        ZABBIX_JS(1)
```

zabbix_proxy

```
ZABBIX_PROXY(8)                                                 ZABBIX_PROXY(8)


       zabbix_proxy -         Zabbix


       zabbix_proxy [-c config-file]
       zabbix_proxy [-c config-file] -R runtime-option
       zabbix_proxy [-c config-file] -T
       zabbix_proxy -h
       zabbix_proxy -V


       zabbix_proxy —       ,
                     Zabbix.


       -c, --config config-file
                               config-file                .

       -f, --foreground
```

Zabbix foreground.

-R, --runtime-control runtime-option
runtime-option.

config_cache_reload
. ,
. Zabbix
Zabbix .
PID
(-c), ,
PID.

snmp_cache_reload
SNMP.

housekeeper_execute
housekeeper. , housekeeper
.

diaginfo[=section]

section . Section historycache,
preprocessing, locks.
.

prof_enable[=target]
, , target

prof_disable[=target]
, , target

log_level_increase[=target]
, ,
target .

log_level_decrease[=target]
, ,
target .

process-type
(availability manager,
browser poller, configuration syncer, data sender, discovery
manager, history syncer, housekeeper, http poller, icmp pinger,
internal poller, ipmi manager, ipmi poller, java poller,
odbc poller, poller, agent poller, http agent poller,
snmp poller, preprocessing manager, preprocessing worker,
self-monitoring, snmp trapper, supervisor, task manager,
trapper, unreachable poller, vmware collector)

process-type,N
(, poller,3)

pid

process-type
(availability manager,
browser poller, configuration syncer, data sender, discovery

```
                manager, history syncer, housekeeper, http poller, icmp pinger,
                internal poller, ipmi manager, ipmi poller, java poller,
                odbc poller, poller, agent poller, http agent poller,
                snmp poller, preprocessing manager, self-monitoring,
                snmp trapper, task manager, trapper, unreachable poller,
                vmware collector)

            process-type,N
                            (     , history syncer,1)

            pid

            scope
                (rwlock, mutex, processing)             process-type
                (     , history syncer,1,processing)

        -T, --test-config
                                    .

        -h, --help
                                .

        -V, --version
                                   .


        /usr/local/etc/zabbix_proxy.conf
                                      Zabbix         (
                                ).

 .


        zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8),
        zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                   ZABBIX_PROXY(8)
```

zabbix_sender

```
ZABBIX_SENDER(1)                                          ZABBIX_SENDER(1)


        zabbix_sender -                 Zabbix


        zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
        host -k key -o value
        zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
        host] [-T] [-N] [-r] -i input-file
        zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
        [-t timeout] [-s host] -k key -o value
        zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
        [-t timeout] [-s host] [-T] [-N] [-r] -i input-file
        zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
        host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
        [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
        cert-subject] --tls-cert-file cert-file --tls-key-file key-file
        [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
        value
        zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
```

```
host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
[-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender -h
zabbix_sender -V


zabbix_sender —
             Zabbix       .        Zabbix
                Zabbix trapper               .
         ,
      ,           «            »
   .


-c, --config config-file
              config-file. zabbix_sender
                         agentd.         zabbix_sender
                         .
       Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile,
       TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject,
       TLSCertFile, TLSKeyFile, TLSPSKIdentity   TLSPSKFile.
       Hostname,               HostnameItem,
          ;
            ( .      -s).
          ,                          ServerActive.
                                            ,
                                    .

-z, --zabbix-server server
                  IP-        Zabbix.
                    ,
               IP-       .
```

--config ServerActive,
agentd.

-p, --port port
, trapper Zabbix.
10051.
--config ServerActive,
agentd.

-I, --source-address IP-address
IP- . --config
SourceIP,
agentd.

-t, --timeout seconds
. : 1-300
(: 60).

-s, --host host
,
(- Zabbix). IP-
DNS- . --config
Hostname,
agentd.

-k, --key key
,
.

-o, --value value
.

-i, --input-file input-file
. -
<input-file>, .
:
<hostname> <key> <value>.
. 3
: <hostname> <key> <value>, "hostname" —
,
- Zabbix, "key" — ,
"value" — . -
<hostname>,
--host.

:

"Linux DB3" db.connections 43

- Zabbix. zabbix_sender 250
.
UTF-8.
, .
:

- .
- .
- , ,
.

- .

- .

- (\n).

- .

-T, --with-timestamps

--input-file.

4
: <hostname> <key> <timestamp> <value>.

Unix. ,
;
.

:

"Linux DB3" db.connections 1429533600 43

. --input-file.

,
« », ;
, ,
.

-N, --with-ns

--with-timestamps.

5
: <hostname> <key> <timestamp> <ns> <value>.

:

"Linux DB3" db.connections 1429533600 7402561 43

. --input-file.

-r, --real-time

.
.

--tls-connect value

. :

unencrypted
( )

psk TLS

cert TLS

```
--tls-ca-file CA-file
                ,
                                  .

--tls-crl-file CRL-file
                ,                    .

--tls-server-cert-issuer cert-issuer
                              .

--tls-server-cert-subject cert-subject
                                 .

--tls-cert-file cert-file
                ,
             .

--tls-key-file key-file
                ,               .

--tls-psk-identity PSK-identity
                   PSK.

--tls-psk-file PSK-file
                ,                          .

--tls-cipher13 cipher-string
                  OpenSSL 1.1.1
       TLS 1.3.
           .               ,        OpenSSL
       1.1.1.

--tls-cipher cipher-string
                  GnuTLS (   TLS 1.2     )
           OpenSSL (       TLS 1.2).
                           .

-v, --verbose
               ; -vv                  .

-h, --help
                                .

-V, --version
                                  .


            0,
               .                ,
                          ,              2.
                        ,              1.


zabbix_sender -c  /etc/zabbix/zabbix_agentd.conf  -k  mysql.queries  -o
342.45
            342.45
       mysql.queries                .
                           Zabbix,
                  .

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host"  -k
mysql.queries -o 342.45
            342.45
```

mysql.queries Monitored Host, Zabbix,
.

zabbix_sender -z 192.168.1.113 -i data_values.txt
data_values.txt Zabbix
IP- 192.168.1.113.
.

echo "- hw.serial.number 1287872261 SQ4321ASDF" | zabbix_sender -c
/usr/local/etc/zabbix_agentd.conf -T -i -

Zabbix, .
,
.

echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113
-p 10000 -i -
Zabbix
IP- 192.168.1.113 10000 .
.

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o
342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file
--tls-cert-file /home/zabbix/zabbix_agentd.crt --tls-key-file
/home/zabbix/zabbix_agentd.key
342.45
mysql.queries Monitored Host IP-
192.168.1.113, TLS .

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o
342.45 --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd"
--tls-psk-file /home/zabbix/zabbix_agentd.psk
342.45
mysql.queries Monitored Host IP-
192.168.1.113, TLS
(PSK).

.

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_server(8),
zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_SENDER(1)

zabbix_server

ZABBIX_SERVER(8) ZABBIX_SERVER(8)

zabbix_server - Zabbix

zabbix_server [-c config-file]
zabbix_server [-c config-file] -R runtime-option
zabbix_server [-c config-file] -T
zabbix_server -h
zabbix_server -V

zabbix_server Zabbix.

-c, --config config-file
                    config-file   .

-f, --foreground
                Zabbix   .

-R, --runtime-control runtime-option
                                   runtime-option.

        config_cache_reload
                              .   ,
                           .   PID-
                                   (
                -c),   ,
             PID-   .

        snmp_cache_reload
                       SNMP.

        housekeeper_execute
                  housekeeper.   ,   housekeeper
                           .

        trigger_housekeeper_execute
                  trigger housekeeper (
).   ,   trigger housekeeper
                            .

        secrets_reload
                           Vault.   ,
                                           .

        diaginfo[=section]

                       .   historycache,
            preprocessing, alerting, lld, valuecache, locks   connector.

                           .

        prof_enable[=target]
                          ,   ,   target

        prof_disable[=target]
                          ,   ,   target

        ha_status
                                       .
                                  .

        ha_remove_node
            ha_remove_node[=name|ID]
                      ID,   ha_status.
                           .

        ha_set_failover_delay[=delay]
                                         .
                                           .
                                      .

```
        log_level_increase[=target]
                              ,               ,
            target

        log_level_decrease[=target]
                              ,               ,
            target


        process-type
                             (alerter, alert manager,
            alert syncer, availability manager, browser poller,
            configuration syncer, configuration syncer worker, connector
            manager, connector worker, discovery manager, escalator,
            ha manager, history poller, history syncer, housekeeper,
            http poller, http agent poller, icmp pinger, internal poller,
            ipmi manager, ipmi poller, java poller, lld manager, lld
            worker, odbc poller, poller, agent poller, preprocessing
            manager, preprocessing worker, proxy poller, proxy group
            manager, report manager, report writer, self-monitoring,
            service manager, snmp poller, snmp trapper, supervisor, task
            manager, timer, trapper, trigger housekeeper, unreachable
            poller, vmware collector)

        process-type,N
                         (     , poller,3)

        pid


        process-type
                             (alerter, alert manager,
            alert syncer, availability manager, browser poller,
            configuration syncer, configuration syncer worker, connector
            manager, connector worker, discovery manager, escalator,
            ha manager, history poller, history syncer, housekeeper,
            http poller, http agent poller, icmp pinger, internal poller,
            ipmi manager, ipmi poller, java poller, lld manager, lld
            worker, odbc poller, poller, agent poller, preprocessing
            manager, proxy poller, proxy group manager, report manager,
            report writer, self-monitoring, service manager, snmp poller,
            snmp trapper, task manager, timer, trapper, trigger
            housekeeper, unreachable poller, vmware collector)

        process-type,N
                         (     , history syncer,1)

        pid

        scope
            (rwlock, mutex, processing)              process-type
            (     , history syncer,1,processing)

-T, --test-config
                               .

-h, --help
                           .

-V, --version
                               .
```

```
       /usr/local/etc/zabbix_server.conf
                                    Zabbix
              (                     ).

 .


       zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)                                  ZABBIX_WEB_SERVICE(8)


       zabbix_web_service —   -    Zabbix


       zabbix_web_service [-c    -  ]
       zabbix_web_service [-c    -  ] -T
       zabbix_web_service -h
       zabbix_web_service -V


       zabbix_web_service —                      -
              Zabbix.


       -c, --config    -
                                                          .

       -T, --test-config
                                   .

       -h, --help
                                        .

       -V, --version
                                   .


       /usr/local/etc/zabbix_web_service.conf
                                          -     Zabbix
              (                          ).




       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06             ZABBIX_WEB_SERVICE(8)
```

Уведомление об авторских правах

Документация Zabbix НЕ распространяется под лицензией AGPL-3.0. Использование документации Zabbix регулируется следующими условиями:

Вы можете создать печатную копию этой документации исключительно для личного использования. Преобразование в другие форматы разрешено при условии, что фактическое содержимое не изменяется и не редактируется каким-либо образом. Вы не должны публиковать или распространять эту документацию в какой-либо форме или на каких-либо носителях, за исключением случаев, когда вы распространяете документацию способом, аналогичным тому, как Zabbix распространяет её (то есть, в электронном виде для загрузки на веб-сайте Zabbix), или на USB-накопителе или аналогичном носителе, при условии, однако, что документация распространяется вместе с программным обеспечением на том же носителе. Любое другое использование, такое как распространение печатных копий или использование этой документации, полностью или частично, в другой публикации, требует предварительного письменного согласия уполномоченного представителя Zabbix. Zabbix оставляет за собой все права на эту документацию, прямо не предоставленные выше.